Socio-Economics Inspired Distributed Systems Design

David Hales
www.davidhales.com
The Open University, UK

Design Group Seminar, Nov. 16th, 2011

Peer-to-Peer (P2P) is not a technology.

It is a (design?) philosophy.

It runs something like this:

Distributed systems are inherently more efficient, robust and responsive to user needs if functionality, where possible, is decentralized. This means that central control, hierarchy and concentration of resources should be avoided, while peer-level coordination should be encouraged.

What are peer-to-peer systems?

- Machines (nodes) on the internet
- Dynamically connecting to a few others
- Cooperating to achieve some task
- So-called “overlay networks”
- Majority of internet bandwidth use is P2P today
- Often associated with illegal copying

Popular applications of P2P

- BitTorrent
  - Open protocol for sharing large files
  - Peers cooperate to speedup downloads
- Skype
  - Closed protocol for voice over IP
  - Peers cooperate to route audio streams
- Bitcoin
  - Open protocol for making payments in virtual currency
  - Peers cooperate to improve security

Big Picture

- P2P can mean "person-to-person" (less technology more person emphasis)
- Social science and economics can inform the design of P2P systems
- P2P systems change economic realities
- New economic models (commons-based peer production – "wealth of networks" by Yochai Benkler)

Big Picture

- What is important?
  - Increasingly these technologies will structure our social interactions
  - When we design them we make social and economic choices
  - We should be aware of these to inform "good design"
  - Design such systems for the common good?
  - What is the common good?
  - How do we design systems for the common good?

What has sociology or economics got to do with peer-to-peer systems?

- P2P systems *are* socio-economic systems
  - Peers cooperate collectively to achieve their goals
  - No peer in the system controls everything
  - Performance results from interactions
  - At the end-of-day users (people) are in control
  - Sociology and economics has studied such phenomena and systems design benefits from this

OK but what use is this to me?

- Knowing some of the economic background should help you to understand:
  - the basic social/economic theory behind P2P
  - how this informs designs
  - how such designs might be improved
  - how to assess new developments and designs
- It is also a fascinating area in itself:
  - If you are interested you can look-up the terms given in *red italics* on Wikipedia for good introductions

Individualism v. Collectivism

- In socio-economic systems individual interests may conflict with collective interests:
  - e.g. over exploitation of a common resource (a river, a field, the atmosphere etc.)
  - e.g. banks - lending (to those who they know can not repay) to gain a commission by selling on the debt to other banks
  - e.g. P2P file sharing system - downloading more than uploading

Individualism v. Collectivism

- Consider a P2P file sharing system:
  - It is in the *collective interest* for all to upload to others so everyone gets the file quickly
  - But it is in the *individual interest* to save bandwidth by only downloading and hence free-riding on others
  - Free-riding (or free-loading) is a perennial problem in P2P file-sharing systems
  - Any efficient system needs to tackle it in some way

The tragedy of the commons

- These kinds of situations have been termed "commons dilemmas" or "common pool resource dilemmas"
- Called "dilemmas" because we would all be better off if we "did the right thing" but there is an individual incentive to do the wrong thing
- G. Hardin (1968) summarized the issue in his famous paper: "The *Tragedy of the Commons*"
- These kinds of situations occur in P2P file-sharing systems like *BitTorrent*

How to avoid the commons tragedy?

- Central enforcement of correct behaviour
  - require centralised agencies and policing
  - ability to identify and track individuals centrally
  - not suitable for pure P2P (but used with private trackers)
- Decentralised methods
  - self-policing producing incentives for cooperation
  - do not require centralised coordination
  - more suitable for pure P2P
  - can apply ideas from "*game theory*"

What is game theory?

- A way to mathematically analyse games assuming we know:
  - number of players
  - possible moves they can make (strategies)
  - outcome of game based on players moves (pay-off)
  - desirability of game outcomes for each player (utility)

What game are you playing?

- Games can be categorised into two types:

1) Zero-sum games
   - when one player wins another loses
   - summing the final utilities of players = 0
   - e.g. poker, chess, monopoly etc.

2) Non-zero-sum games
   - utilities do not always sum to zero
   - both players may lose or both may win
   - considered to capture social / economic realities
   - e.g. tragedy of the commons examples

Capturing a commons tragedy with a simple game

- Consider a game composed of two players:
  - each player:
    - has choice of one move (C or D)
    - makes a single move then the game ends
    - does not know how the other will move
    - gets a payoff (or utility) based on how they moved and how the other player moved
  - for certain payoff values this game can, minimally, capture a form of commons tragedy (or dilemma)
  - a classic such game is called the *Prisoner's Dilemma*

The Prisoner's Dilemma - "payoff matrix"

Game is a PD when: T > R > P > S and 2R > T + S

Player 1 / Player 2	C	D
C	(3) R R (3)	(5) T S (0)
D	(0) S T (5)	(1) P P (1)

The Prisoner's Dilemma - example games

Players =>	P1	P2	P1	P2	P1	P2	P1	P2
Moves =>	C	C	C	D	D	C	D	D
Payoffs =>	R	R	S	T	T	S	P	P
Values =>	3	3	0	5	5	0	1	1
Total =>	6		5		5		2	

A contradiction between collective and individual interests

Game theory says defect!

- Game theory assumes players are:
  - rational - attempt to maximise *their* utility
  - selfish - don't care about the other guy
  - knowledgeable - have complete information
  - clever - have unlimited computational time
- Given these assumptions it can be proved:
  - agents will select equilibria where no player will improve by changing strategy unilaterally
  - many games have such equilibria - by the famous John Nash (so-called *Nash Equilibrium* - NE)
  - the NE for the PD is DD (all defect)

Iterated Prisoner’s Dilemma

- Previous example “one-shot” PD but:
  - real world interactions often repeated
  - might meet the guy you just ripped-off in the future
  - allows for more complex sequence of strategies based on past interactions with others
  - can punish someone tomorrow for defecting against you today - “the shadow of the future”
- Iterated PD (IPD) captures this and, as we will see, maps well onto P2P file-sharing protocols like BitTorrent

What is the rational thing to do in the IPD?

- Traditional game theory has trouble here:
  - cooperative equilibria exist in infinitely repeated games but not in finite games of known length
  - many equilibria exist and it is not clear which one would be chosen by rational agents
  - In all cases defection on every round is still a equilibrium even when cooperative equilibria exist
- For these reasons *Robert Axelrod* (political scientist), in the late 70's, decided to find out what kinds of strategies worked well in the IPD by using computer simulation

Axelrod’s Tournament - programs as strategies

- Axelrod organised an open IPD tournament:
  - Academics were asked to submit programs (BASIC or FORTRAN) that would play the IPD against each other
  - Nobody knew competitors code
  - The only input would be the on-going past history of the game (a string of C’s and D’s)
  - The aim was to get the highest score (utility) based on round-robin playoffs between all pairs of programs
  - Axelrod’s aim was to see which programs did best against all the others and understand why
  - He wrote-up his results in the famous book “the evolution of cooperation”

Axlerod's Tournament - what happened?

- Basic results were:
  - many strategies were submitted (complex and simple)
  - the one with the highest overall score turned out to be simple: *tit-for-tat* (TFT) or "look back"
  - starts playing C, then "looked back" at the last move made by opponent and copied that move
  - submitted by Psychologist Anatol Rapoport
  - didn't "win" against each strategy but did better overall on average against all strategies
  - TFT mechanism an example of "*reciprocal altruism*" (Robert Trivers)

What has this got to do with BitTorrent?

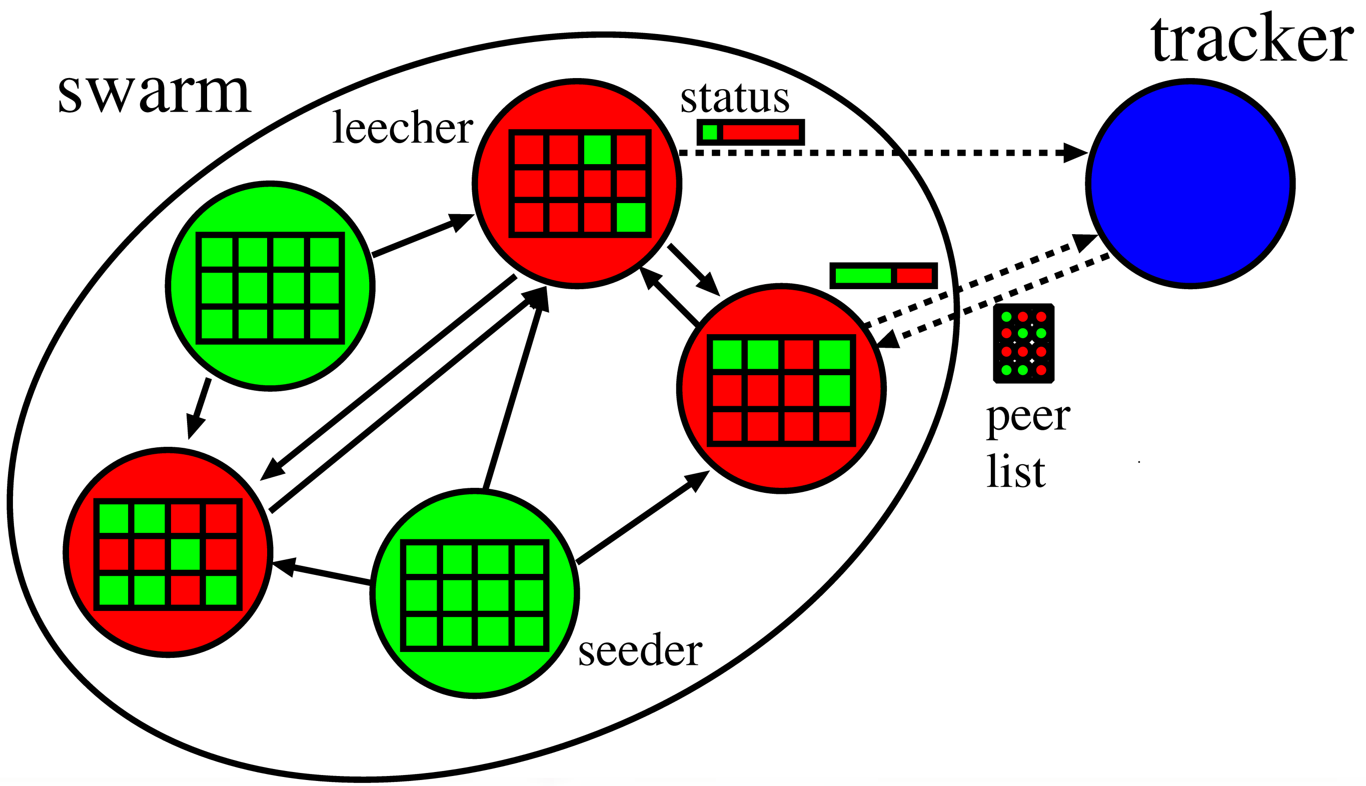
- In the *BitTorrent protocol*:
  - TFT-like method used for sharing files
  - nodes form groups interested in a particular file (swarms) and swap or "barter" pieces with each other
  - if a node does not upload data then this can be compared to playing defection
  - it is punished in the future by being "choked" - not getting upload from others
  - even if you hack your client to be selfish the chances are the standard TFT-like protocol will do better overall
  - *Bram Cohen* - original BT designer - inspired by Axelrod's tournaments

Some BitTorrent Terminology

- Swarm: set of peers interested in a file
  - file is split in smaller chunks called pieces
  - seeder: holds a full copy of the data
  - leecher: holds only a part of the data (initially nothing)
- Tracker: centralized manager
  - keep track of all peers in the swarm
  - return list of current peers in swarm
- Torrent file: meta-data
  - contains pointer to tracker hosting the swarm
  - details about the file - hash, no. of pieces, size etc.

BitTorrent Protocol

- Get a list of other peers in the swarm from the tracker
- Ask peers their list of pieces and tell them what is yours
- Exchange pieces with appropriate peers



The Global Ecology of BitTorrent Clients

- Many *bittorrent clients* exist in "the wild"
  - Bittorrent 6 (from Bittorrent.com, formally utorrent)
  - Others: Azureus, ABC, Transmission, many others…
  - bad guy clients: BitThief, BitTyrant
- Hence:
  - The current bittorrent ecosystem is a ***global on-going experiment***, like Axelrod's, but with huge user base and rich interactions (not just TFT) incredible strategy sophistication
  - This is unprecedented and will surely lead to new economic theory - in general!

BitTorrent Clients

BitTorrent client	FOSS	Linux/Unix	Windows	Mac OS X	IPv6[1]	Programming language	Based on	Interface	Spyware/Adware /Malware-free
ABC	Yes	Partial	Yes	No	buggy[2]	Python	BitTornado	GUI and web	Yes
Acquisition	No	No	No	Yes	?	Objective-C and Cocoa	Limewire	GUI	Yes
Anatomic P2P	Yes	Yes	Yes	Yes	No	Python	BitTornado	GUI and old CLI	Yes
Arctic Torrent	Yes	No	Yes	No	No	C++	libtorrent	GUI	Yes
aria2	Yes	Yes	Yes	Yes	?	C++	-	CLI	Yes
Azureus	Yes	Yes	Yes	Yes	Partial[3]	Java and SWT	-	GUI, CLI, Telnet, Web, XML over HTTP remote control API	Yes
BitComet	No	No	Yes	No	No	C++	?	GUI	Yes[4]
BitFlu	Yes	Yes	No	Yes	Yes	Perl	-	Telnet and Web	Yes
BitLet	Planned	Yes	Yes	Yes	?	Java and JavaScript	-	Web XHTML	Yes
BitLord	No	No	Yes	No	No	C++	BitComet	GUI	Adware
BitPump	No	No	Yes	No	No	C++	-	GUI	Yes
Bits on Wheels	No	No	No	Yes	No	Objective-C and Cocoa	-	GUI	Yes
BitSpirit	No	No	Yes	No	No	C++	BitComet	GUI	Yes
BitThief	No	Yes	Yes	Yes	?	Java	?	GUI	Yes
BitTornado	Yes	Yes	Yes	Yes	Yes	Python	BitTorrent	GUI and CLI	Yes
BitTorrent 5 / Mainline	Yes	Yes	Yes	Old version	No	Python	-	GUI and CLI	Yes
BitTorrent 6	No	No	Yes	No	Yes	C++	µTorrent	GUI and CLI	Yes
BitTyrant	Yes	Yes	Yes	Yes	Partial[3]	Java and SWT	Azureus	GUI, CLI, Telnet, Web, XML over HTTP remote control API	Yes
Blizzard Downloader	No	No	Yes	Yes	?	?	BitTorrent client for early version	GUI	Yes
Blog Torrent	Yes	No	Yes	Yes	?	?	BitTorrent client for early version	GUI	Malware-Status: unknown
BTG	Yes	Yes	Partial[5]	Yes	No	C++	libtorrent	CLI, GUI and web	Yes

Take home message

- Previous work in social / economic science (Axelrod's IPD) has provided a basis for protocol design in a P2P system
- Deployed variants of the protocol are creating a massive global economic experiment
- Measurements can be made and these could inform new theory and new protocols

References

- Nash, John (1950) Equilibrium points in n-person games. Proceedings of the National Academy of Sciences 36(1):48-49.
- John von Neumann and Oskar Morgenstern: Theory of Games and Economic Behavior, Princeton University Press (1944)
- Robert Axelrod (1984) The Evolution of Cooperation, Basic Books
- Nowak, M.A. and Sigmund, K. (1998) Evolution of indirect reciprocity by image scoring, Nature 393, 573.
- Garrett Hardin (1968) The Tragedy of the Commons Science 162, 1243-1248.
- Trivers, R.L. (1971). The evolution of reciprocal altruism. Quarterly Review of Biology. 46: 35-57
- Maynard Smith, J. (1982) Evolution and the Theory of Games. Cambridge University Press

work using socio-economic ideas

- Effort based incentive approaches from participatory economics applied in BT
  – Rahman, R., Meulpolder, M., Hales, D., Pouwelse, J. and Sips, H. (2010) Improving Efficiency and Fairness in P2P Systems with Effort-Based Incentives. Proceedings of the IEEE International Conference on Communications, 23-27th May 2010, Cape Town, South Africa
- Analysis of credit shortages and “monetary policy” in private BT communities
  – Hales, D., Rahman, R., Zhang, B., Meulpolder M., and Pouwelse, J. (2009) BitTorrent or BitCrunch: Evidence of a credit squeeze in BitTorrent? Proceedings of the 5th Collaborative Peer-to-Peer Systems (COPS) Workshop, in conjunction with 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, June 29 - July 1, 2009, Groningen, the Netherlands.
  – Rahman, R. and Hales, D., Vinko, T., Pouwelse, J. and Sips, H. (2010). No more crash or crunch: sustainable credit dynamics in a P2P community. International Conference on High Performance Computing & Simulation (HPCS 2010), Caen, France, 2010.
- Apply Axelrod-like tournaments to realistic BT protocol
  – Joint work with Rameez Rahman, Tamás Vinko, David Hales, Johan Pouwelse, Henk Sips (2011) Design Space Analysis for Modeling Incentives in Distributed Systems, to be presented at Sigcomm August 2011, Toronto.

Design Space Analysis for Modelling Incentives in Distributed Systems

- Mainly thesis work of Rameez Rahman
- Apply Axelrod-like tournament approach to evaluate *realistic* P2P protocol variants
- Interesting bit is:
  - break down of P2P protocols into a design space
  - Evaluation of protocol variants (PRA)
- Specific application to BitTorrent protocol variants

PRA characterisation of a protocol

- ***Performance*** - the overall performance of the system when all peers execute Π (where performance is determined by the designer);
- ***Robustness*** - the ability of a majority of the population executing Π to outperform a minority executing a protocol other than Π;
- ***Aggressiveness*** - the ability of a minority of the population executing Π to outperform a majority executing a protocol other than Π.

More detail on PRA

- P = average download time
- R = number of "wins" in round robin tournaments against all other protocol variants
- A = number of "wins" in round robin tournaments against all other protocol variants
- P,R,A values normalised over the space

Parameterising a P2P protocol

- **Peer Discovery**: In order to perform productive peer interactions, it is necessary to find other partners. For example, when a peer is new in the system, looking for better matching partners or existing partners are unresponsive. The timing and nature of the peer discovery policy are the important aspects of this dimension.
- **Stranger Policy**: When interacting with an unknown peer (stranger), past history cannot be used to inform actions. It is therefore necessary to apply a policy to deal with strangers. The way peers allocate resources to strangers is an important aspect of this dimension.
- **Selection Function**: When a peer requires interaction with others this function determines which of the known peers should be selected. This could include, for example, past behaviour (through direct experience or reputation system), service availability and liveness criteria.
- **Resource Allocation**: During peer interactions resources must be allocated to the selected peers (given by the selection function). The way a peer divides its resources among the selected peers, defines the resource allocation policy.

Parameterising BT

- Stranger policy (10 variants)
- Selection function:
  - Candidate list - peers to consider (2 variants)
  - Ranking function - order list (6 variants)
  - Selection - number of peers to select (9 variants)
- Resource allocation (3 variants)
- Gives a space of 3270 unique protocols

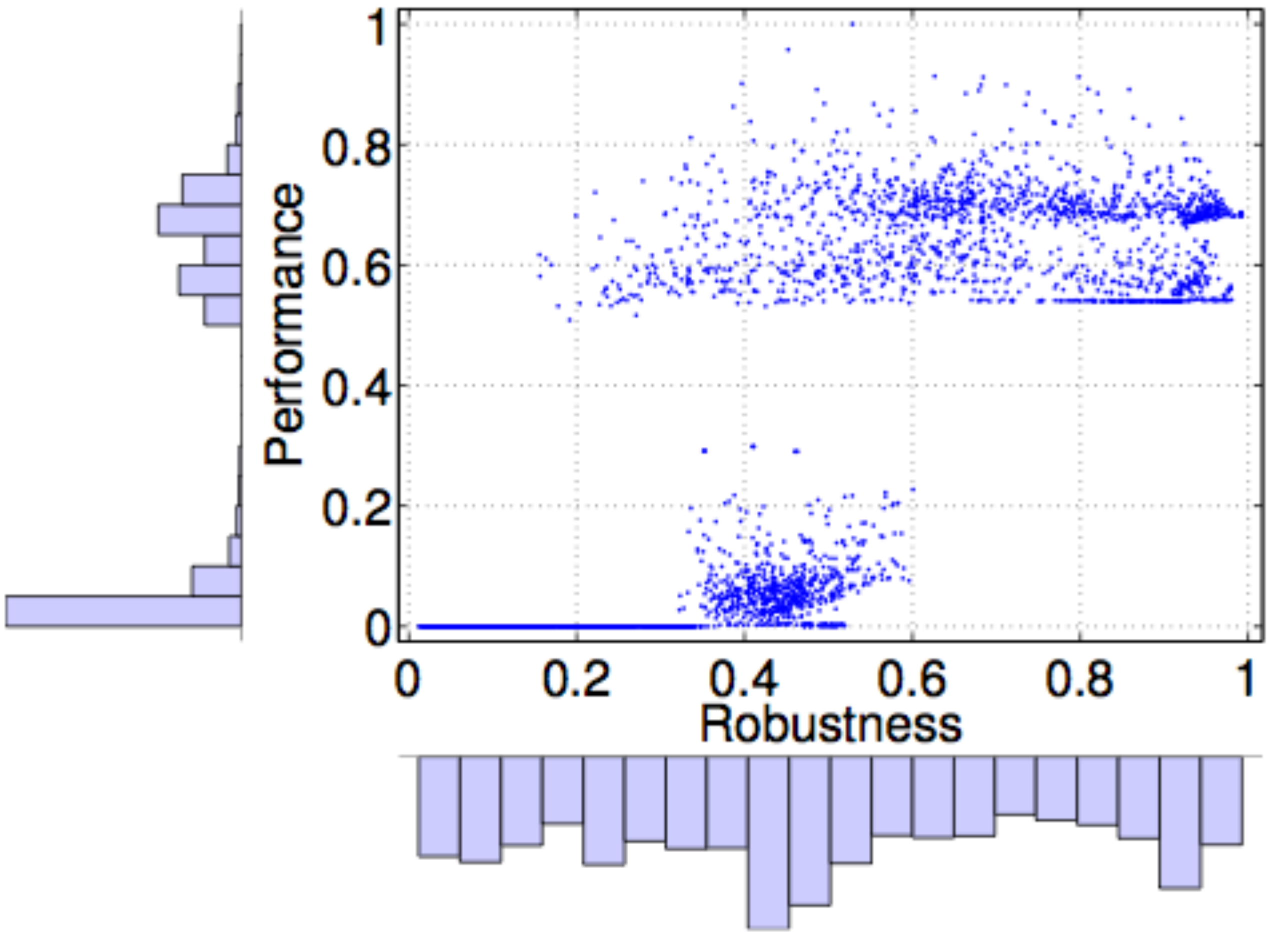


Figure 3: Scatter plot of all 3270 protocols in the design space with Robustness against Performance. The results presented here are a synthesis of over 107 million individual simulation runs. Histograms are also shown.

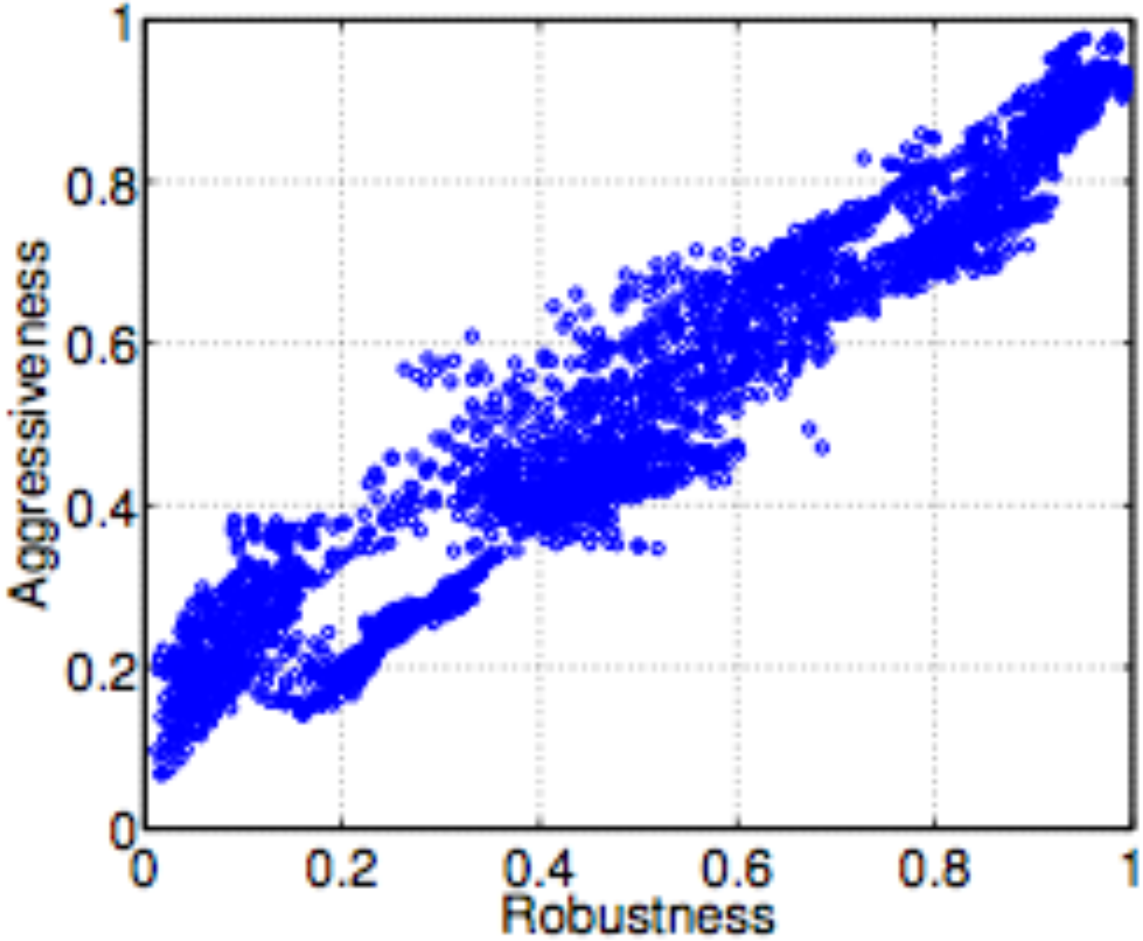


Figure 9: Scatter plot of robustness and aggressiveness values of the protocols. The Pearson's correlation coefficient is 0.96.

Broad Summary

- Lower cluster (low P) all free rider variants who do not reciprocate with partners
- Upper cluster (high P) do reciprocate with partners but some defect with strangers
- Top P, low number of partners (1,2), Sort Loyal, When Needed
- Top R, high number of partners (6-9), Sort Fastest, When Needed, Prop. Share
- Sweet spot (P,R>0.8): Sort Loyal

Interesting link to some economic work?

- ***Compare empirical / modelling work***: Kirman AP and Vriend NJ (2000) "Learning to be loyal: A study of the Marseille fish market" In: Gatti DD, Gallegati G and Kirman AP, Interaction and market structure: essays on heterogeneity in economics, Volume 484. Springer,

From swarms to collectives

Where we start to see things that look a bit like “real economics” emerge

Communities have formed around BitTorrent Trackers

Quite a few of these:

Linux tracker

Public Trackers (e.g. PirateBay)

- BitTorrent uses Trackers to index swarms
- Public trackers let anyone join or create a swarm
- Sharing within a swarm is incentivised via a form of tit-for-tat (as we have seen)
- However there is no incentive for:
  - Seeding (uploading after file is downloaded)
  - Capping (creating and injecting a new file)
  - Maintaining a Tracker in the first instance

Private Trackers (Many)

- Private Trackers have emerged more recently
- Only allow registered users to join swarms
- May track upload / download of each user
- Some keep centralised accounts for each user
  - When users download much more than upload they may be kicked out
  - Many different schemes: ratio, credits, points etc
- Some rely on users to just be nice with various "gentleman's club" methods

A little detail on credit systems

- We will give a little detail on credit systems in private BT communities
- Give a flavour of how economic / collective issues are becoming significant
- Present results from a simple (agent-based) model and some measurements of a real private tracker

Private Trackers - Credit

- Consider a scheme based on credits
  - Uploading 1MB earns one credit
  - Downloading 1MB costs one credit
  - A user with no credits can't download
- Users must be given some initial credit
- In fixed size pop. total credit remains constant
- Similar to a fixed supply of money in an economy (loose analogy!)

Private Trackers - Credit

- How much credit should be put into the system?
- How would it effect the efficiency of the system?
- When do credit squeezes occur?
- How can they be avoided?

We define a credit squeeze as a situation in which, due to lack of credit, the efficiency of the system is significantly reduced.

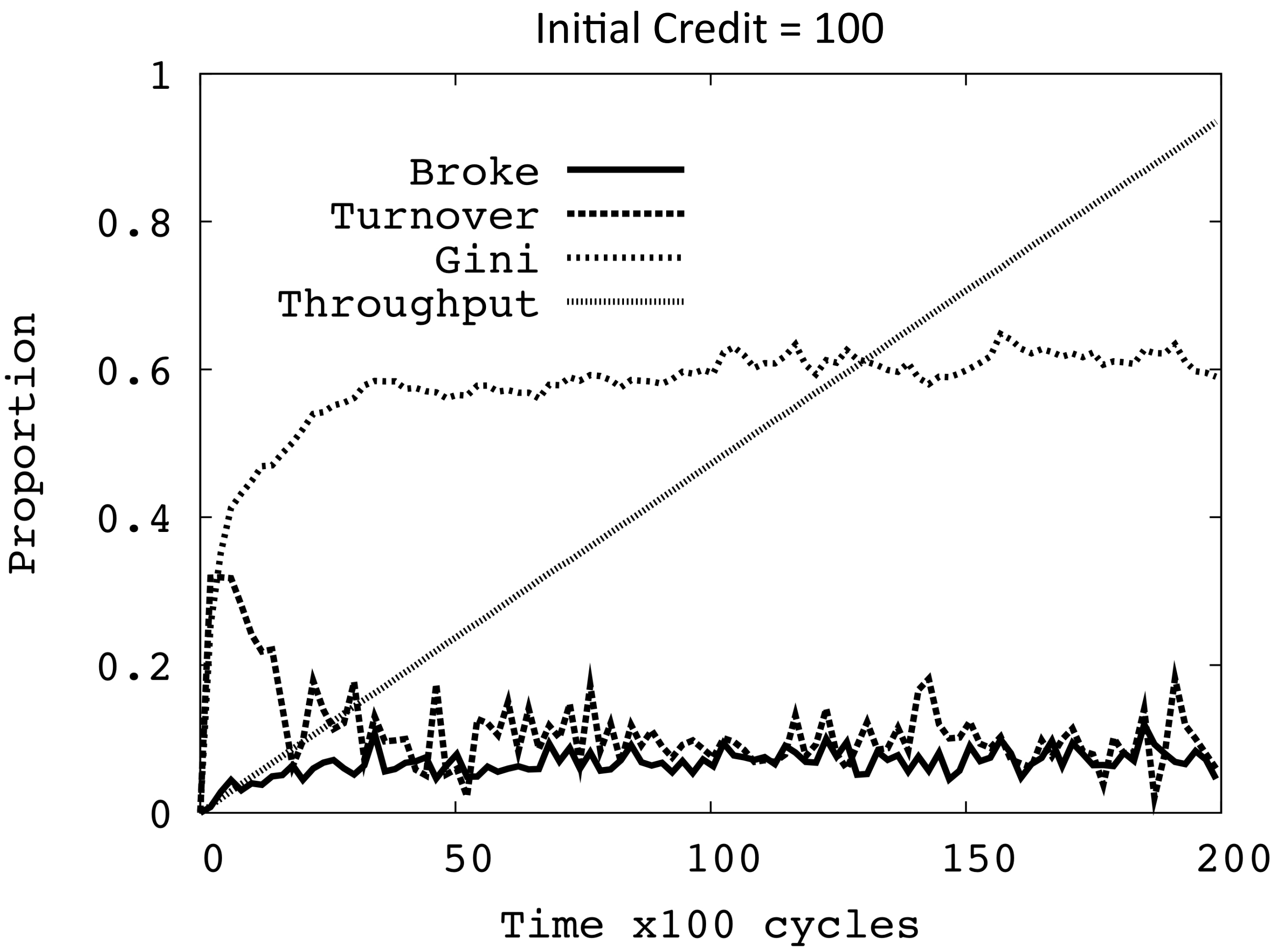
BitCrunch Model

- Highly abstract and simplified model
  - All nodes have equal upload / download
  - Equally interested in all swarms on the tracker
  - Always uploading to one swarm (seeding)
  - Always downloading from another swarm (leeching)
  - No modelling of tit-for-tat or free-riding
  - Always online, fixed population
  - If run out of credit (broke) must wait until earns some via upload before being allowed to download
  - Swarms assumed to share upload “perfectly”

BitCrunch Model – baseline runs

- Parameters:
  - 500 peers, 100 swarms
  - Peer upload and download capacity = 1 unit
  - Each file shared in each swarm = 10 units size
  - One simulation cycle = each swarm processes one unit of time
  - Run for 20,000 cycles (x10 runs)
  - For initial credit per peer of 1, 10 and 100 units

Typical basline simulation run



Baseline simulation results

C	T	β	G	φ
1	0.58	0.36	0.87	0.84
10	0.81	0.20	0.77	0.43
100	0.97	0.06	0.59	0.10

C = initial credit

T = total throughput = total number of units uploaded as proportion of maximum possible (infinite credit)

B = proportion of nodes that are "broke" (zero credit)

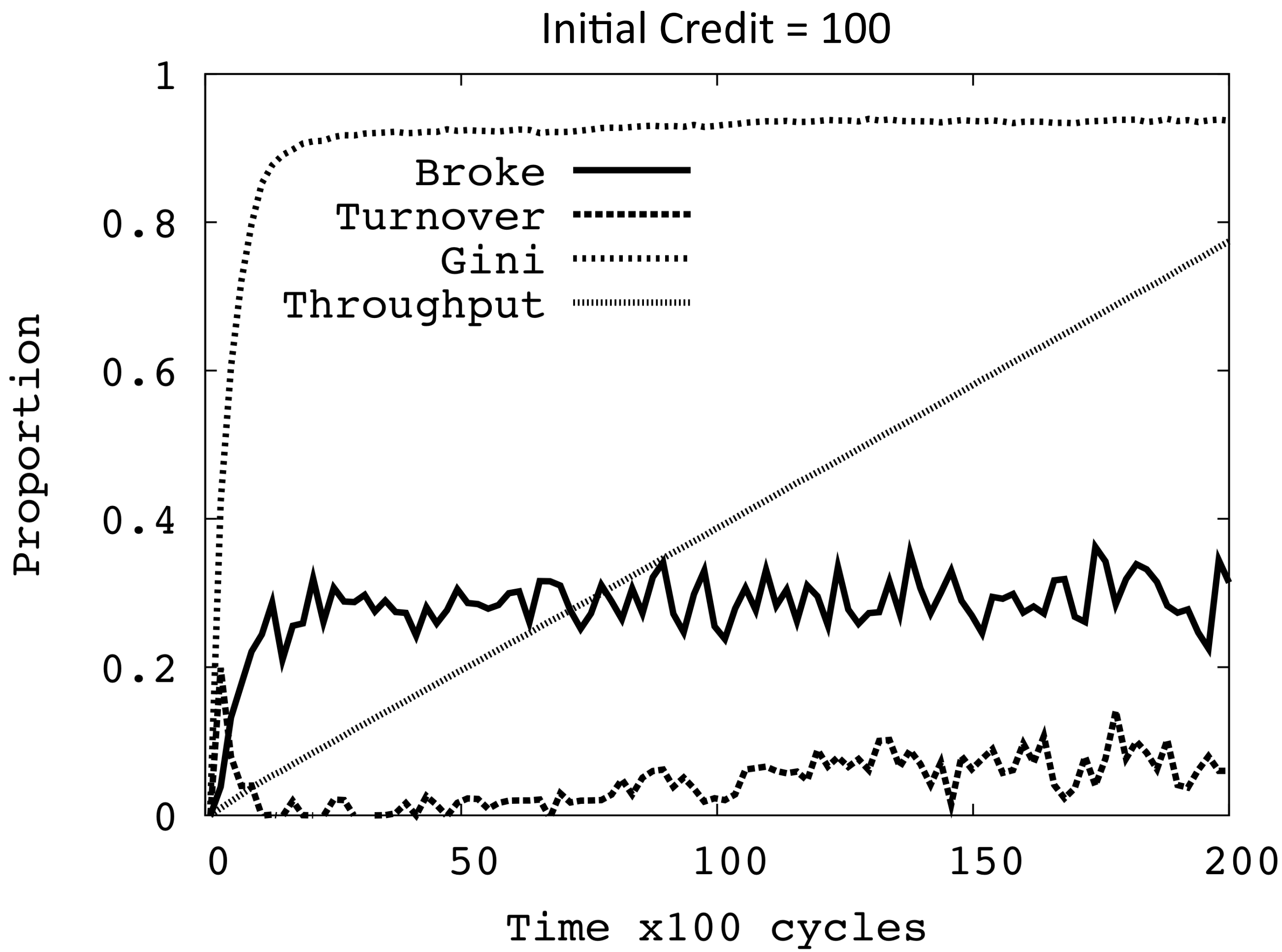
G = Gini measure (simple measure of inequality of credit)

Phi = turnover of top 10% of peers ranked by credit (credit mobility)

Unequal capacities runs

- To determine what happens when some nodes of different upload capacities
- Parameters (same as baseline runs but):
  - All peers download capacity = 10 units
  - 10% of peers upload capacity = 10 units
  - 90% of peers upload capacity = 1 unit
  - Examined a (1.5 credit) seeding bonus approach to dynamically introduce more credit into the system

Typical unequal capacities run



Unequal capacities simulation results

C	T	β	G	φ
1	0.56	0.39	0.90	0.82
10	0.71	0.32	0.93	0.44
100	0.77	0.29	0.94	**0.06**
100++	0.97	0.01	0.71	0.00

C = initial credit

T = total throughput = total number of units uploaded as proportion of maximum possible (infinite credit)

B = proportion of nodes that are "broke" (zero credit)

G = Gini measure (simple measure of inequality of credit)

Phi = turnover of top 10% of peers ranked by credit (credit mobility)

100++ indicates initial credit of 100 with 1.5 credit seeding bonus

Observations

Even in a trivial model where all peers have the same capacities and user behaviour, all swarms have equal popularity and all peers start with equal credits, the performance of the system may be inhibited by credit shortages

Observations

Adding extra capacity to the system, in the form of upload and download, can actually reduce the performance. This is highly counter intuitive and something that should be avoided because it implies lack of scalability.

Observations

By injecting new credit into the system in the form of a ``seeding bonus'' a credit squeeze can be ameliorated when peer capacities are unbalanced.

Statistics from a Private Tracker

Day	T	Δ	Δ_0	δ	S/L
1	48	24	17	0.23	26
2	40	20	15	0.25	26
3	50	25	12	0.16	25
4	67	33.5	17	0.17	25
5	52	26	19	0.24	25
6	46	23	15	0.21	25
7	87	43.5	17	0.13	25
Ave.	56	28	16	0.19	25

Approx. 50,000 peers per day, 10,000 swarms,
access to credit balances of top 10%

T = throughput in TB over all swarms
Delta = total credit increase that day in the entire system
Delta0 = total credit increase for top 10% of peers
Delta = minimum fraction of credit increase that goes to top 10% of peers
S/L = seeder to leecher ratio over all swarms

Statistics from a Private Tracker

- Indicates “rich getting richer” since top 10% are getting a lot of the new credit
- High Seeder / Leecher ratio suggestive that a credit squeeze is happening for many
- But need more information to verify this
- Would be interesting to see what happened to throughput if there was a “free day” or seeding bonus was increased

Conclusions

- Private trackers using “ratio enforcement” policies appear to be ad-hoc and various
- But can have dramatic effects on efficiency
- Too much credit could encourage free-riding
- Too little creates squeezes = lower efficiency
- These are just initial investigations
- Much more work needs to be done!

Take home message

- Communities formed around trackers provide an on-going global socio-economic experiment
- Self-organisation of socio-economic structures in measurable forms
- Ideas, models and theories from socio-economics may inform and learn from this
- Such communities so strong don't be surprised if they start influencing the "real world" (e.g. the PirateParty)

References

- Rahman, R. and Hales, D., Vinko, T., Pouwelse, J. and Sips, H. (2010). **No more crash or crunch: sustainable credit dynamics in a P2P community**. International Conference on High Performance Computing & Simulation (HPCS 2010)
- Rahman, R., Meulpolder, M., Hales, D., Pouwelse, J. and Sips, H. (2010) **Improving Efficiency and Fairness in P2P Systems with Effort-Based Incentives**. Proceedings of the IEEE International Conference on Communications
- Hales, D., Rahman, R., Zhang, B., Meulpolder M., and Pouwelse, J. (2009) **BitTorrent or BitCrunch: Evidence of a credit squeeze in BitTorrent?** Proceedings of the 5th Collaborative Peer-to-Peer Systems (COPS) Workshop

P2P Open Source Currency

- Similar to file sharing systems (like BitTorrent)
- Fully decentralized no central trusted authority
- Collectively “owned” and “controlled” by users
- Cryptography provides security and privacy
- Open source code provides transparency
- Large design space of monetary policies possible

Example 1: BitCoin (bitcoin.org)

- Analogous to “gold” – policy:
  - There are a maximum number of bitcoins that can exist 21 million (8 decimal places granularity)
  - Produced through “mining” CPU intensive search increases over time
- All transactions broadcast to all nodes
- If majority of CPU power in network runs protocol cheating is hard
- Mining and potential transaction fees incentivise “good behavior”
- Several exchanges, few traders accept, market CAP = $44m

Bitcoin WATCH

Latest News View All

Mon 2:00 SC Magazine: BitCoin forum hack
Mon 1:30 Present Cynosure: Krugman's Bitcoin Error
Thu 21:00 Latest BitLotto Winner: "Where Did All These Bitcoins Come From?"
Wed 20:45 Michael Suede - Questions For Krugman
Wed 10:30 Bitcoin-Trader - Interview Series Part 3: David Sterry - ExchB
Wed 9:45 Paul Krugman - Golden Cyberfetters [Op-Ed on Bitcoin]

Home | Technical Analysis | Market Overview

Economy

Total BTC	7,251,000 BTC
Market Cap based on latest prices	44,075,929 USD or 38,067,750 EUR or 143,473,507 PLN or 29,004,000 GBP
Transactions last 24h	6,685
Transactions avg. per hour	278.54
Bitcoins sent last 24h	1,105,541.27 BTC
Bitcoins sent avg. per hour	46,064.22 BTC

Exchanges

Market	Last	Volume (24h)	Bid	Ask	High	Low
mtgoxUSD	6.2292	164,597.61	6.20	6.20	12.50	3.85
thUSD	6.0786	6,723.39	6.09	6.10	6.89	4.51
mtgoxRUB	133.8124	2,500.00	184.75	189.43	141.08	88.43
btceUSD	6.0000	1,531.83	5.90	6.01	6.66	4.70
bitchangePLN	18.0000	1,314.06	17.60	18.00	18.34	15.70
virtexCAD	6.4990	1,163.21	6.00	6.99	7.10	5.40
britcoinGBP	4.0000	998.03	3.86	3.95	4.26	2.96
mtgoxPLN	19.7867	619.90	19.10	19.87	22.00	14.48
mtgoxEUR	4.5494	589.58	4.55	4.60	5.01	3.30
exchbUSD	6.3000	485.95	6.20	6.34	7.45	4.99
virwoxSLL	1829.0000	448.00	1700.00	1828.00	1830.00	1283.00

Example 2: Ripple (Ripplepay.com)

- Analogous to "bank credit" – policy:
  - Anyone in the system can issue credit
  - But only to trusted friends (social network)
  - Several units of account (incl. bitcoin)
- Security and incentives based on existing trust
- Early stage no full P2P implementation
- No traders or exchanges

Problems…

- No single policy or software implementation will work for all people, time and places
- But given a sufficient ecology of competing systems…
- New systems can emerge, hacked or less useful systems will dissolve
- Hacked systems introduce a new kind of Gresham's law
- Competition between systems can drive cooperation within them: group selection

Problems…

- Each P2P currency is an island (rather like a nation) in which value is trapped
- Requires trusted 3rd parties to provide exchanges to allow for movement of value
- Fully distributed P2P exchanges?
- Some attempt:

  https://github.com/macourtney/Dark-Exchange

Solutions?

- Assuming open source P2P:
  - Healthy ecology of completing currencies
  - Liquid exchanges
- Could we borrow an idea from the Chinese central bank?
- A composite currency based on a basket of popular P2P currencies
- Using open source algorithm to constantly rebalance basket to maintain value

Agent-based models?

- Group selection models may be adapted where:
  - Group = agents holding currency
  - Cooperation = behaviour that maintains value of currency
  - Defection = behaviour that inflates or deflates value (including hacking)
  - Under assumption that agents are boundedly rational copiers of others

A “sparkling” economy

- Even an ecology of constantly forming and bursting bubbles “a sparkling economy”
- Might produce stable value given a sufficiently “cleaver” algorithm
- But no algorithm can know the future for sure
- Could we evolve them in simulation?
- A task for NESS?

Questions?

Some References:

[1] John F. Nash, “Ideal Money”, Southern Economic Journal, 2002, 69 (1). http://www.jstor.org/pss/1061553

[2] H. G. Wells, “The New World Order”, 1940. http://gutenberg.net.au/ebooks04/0400671h.html

[3] People's Bank of China, "Reform the Int. Monetary System", March 2009. http://is.gd/0OhCmS

[4] F. A. Hayek, “The Denationalisation of Money: The Argument Refined”, 1990. http://mises.org/books/denationalisation.pdf

[5] The Economist, Jun 13th 2011. http://www.economist.com/blogs/babbage/2011/06/virtual-currency

[6] Victor Grishchenko, “Bitcoin?”, May 12th 2011. http://www.pds.ewi.tudelft.nl/~victor/bitcoin.html

[7] Gresham’s Law - http://en.wikipedia.org/wiki/Gresham's_law

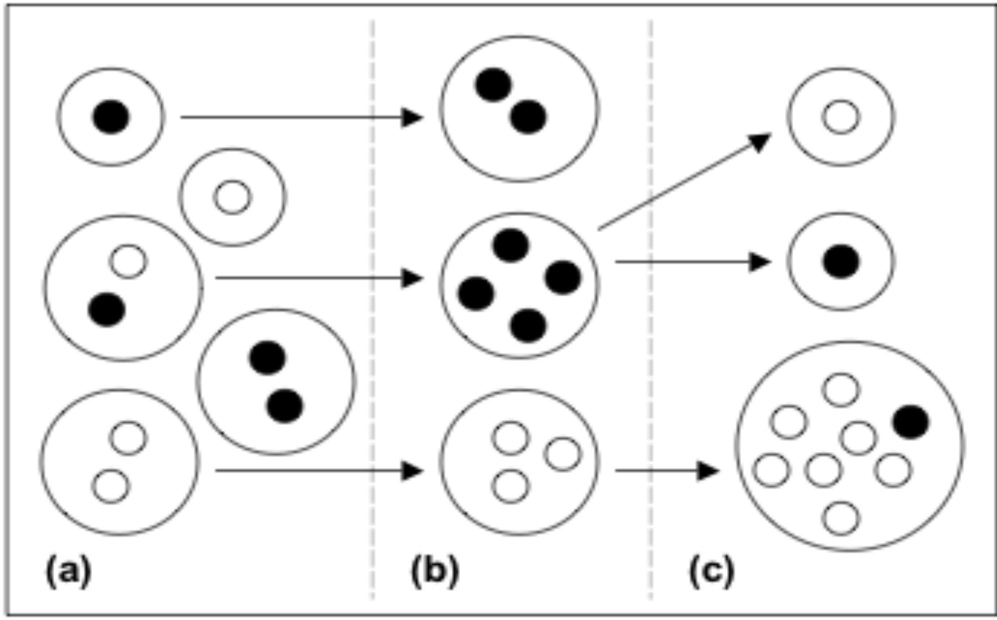
New Group Selection Models

Group Selection Models

- Recent models of "group selection"
- Based on individual selection
- Producing dynamic social structures
- Limit free-riding
- Increasingly group-level performance
- Don't require reciprocity
- Could be very useful in P2P

Evolutionary Group Selection Models

- *Group boundary* - a mechanism which restricts interactions between agents such that the population is partitioned into groups
- *Group formation* - a process which forms groups dynamically in the population
- *Migration* - a process by which agents may move between different groups
- *Conditions* - cost / benefit ratio of individual interactions and other conditions which are sufficient for producing group-level selection



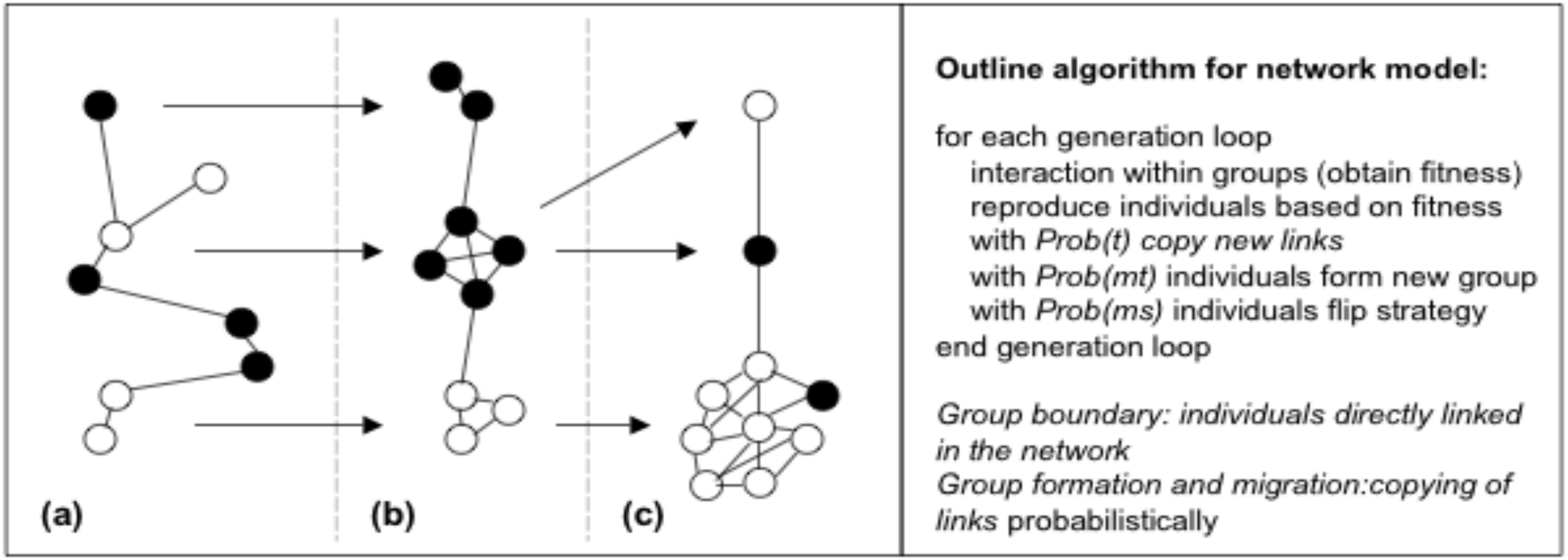
Outline algorithm for tag model:

for each generation loop
  interaction within groups (obtain fitness)
  reproduce individuals based on fitness
  with *Prob(mt)* individuals form new group
  with *Prob(ms)* individuals flip strategy
end generation loop

Group boundary: tag stored by each individual defines group membership
Group formation and migration: probabilistic mutation of tag

Schematic of the evolution of groups in the tag model. Three generations (a-c) are shown. White individuals are pro-social (altruistic), black are selfish. Individuals sharing the same tag are shown clustered and bounded by large circles. Arrows indicate group linage. When ***b*** is the benefit a pro-social agent can confer on another and ***c*** is the cost to that agent then the condition for group selection of pro-social groups is: ***b > c and mt >> ms***

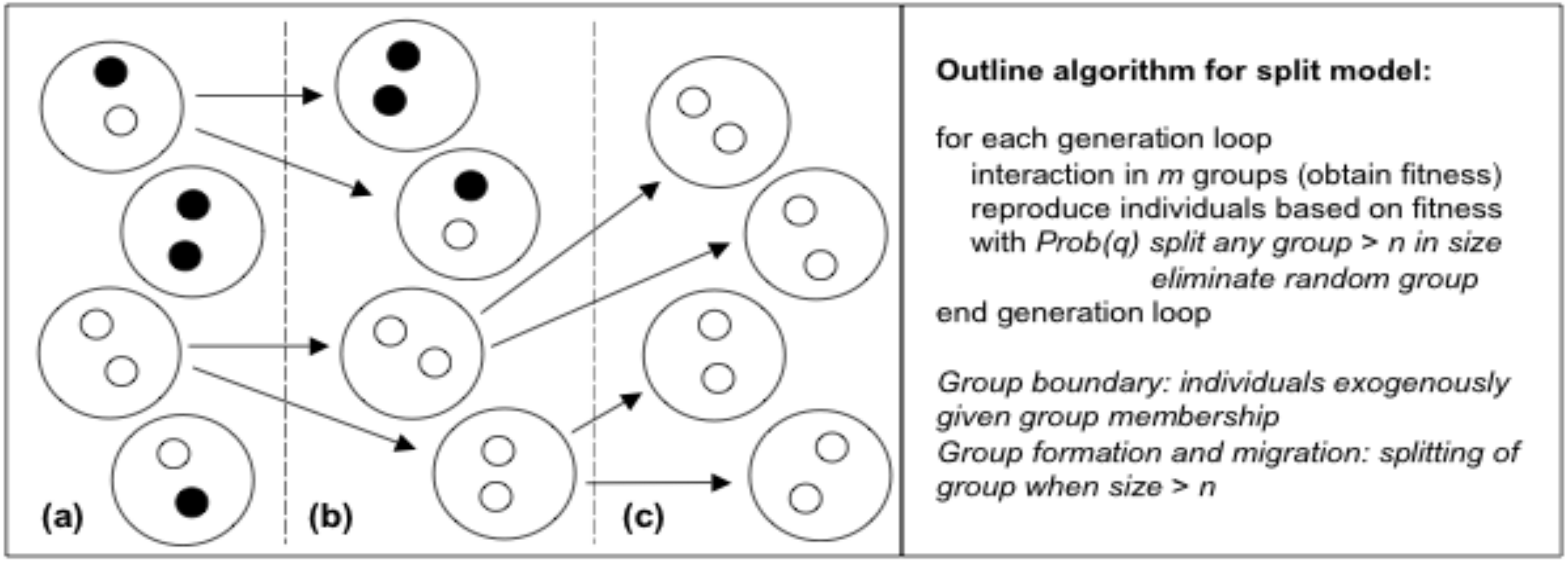
Riolo, Axelrod, Cohen, Holland, Hales, Edmonds…



Schematic of the evolution of groups in the network-rewire model. Three generations (a-c) are shown. Altruism selected when:***b > c and mt >> ms.*** When ***t = 1***, get disconnected components, when ***1 > t > 0.5***, get small-world networks

Hales, D. & Arteconi, S. (2006) Article: SLACER: A Self-Organizing Protocol for Coordination in P2P Networks. IEEE Intelligent Systems, 21(2):29-35

Santos F. C., Pacheco J. M., Lenaerts T. (2006) Cooperation prevails when individuals adjust their social ties. PLoS Comput Biol 2(10)



Schematic of the evolution of in the group-splitting model. Three generations (a-c) are shown. Altruism is selected if the population is partitioned into ***m*** groups of maximum size ***n*** and ***b / c > 1 + n / m.***

Traulsen, A. & Nowak, M. A. (2006). Evolution of cooperation by multilevel selection. Proceedings of the National Academy of Sciences 130(29): 10952-10955.

SLAC: Network re-wire P2P model

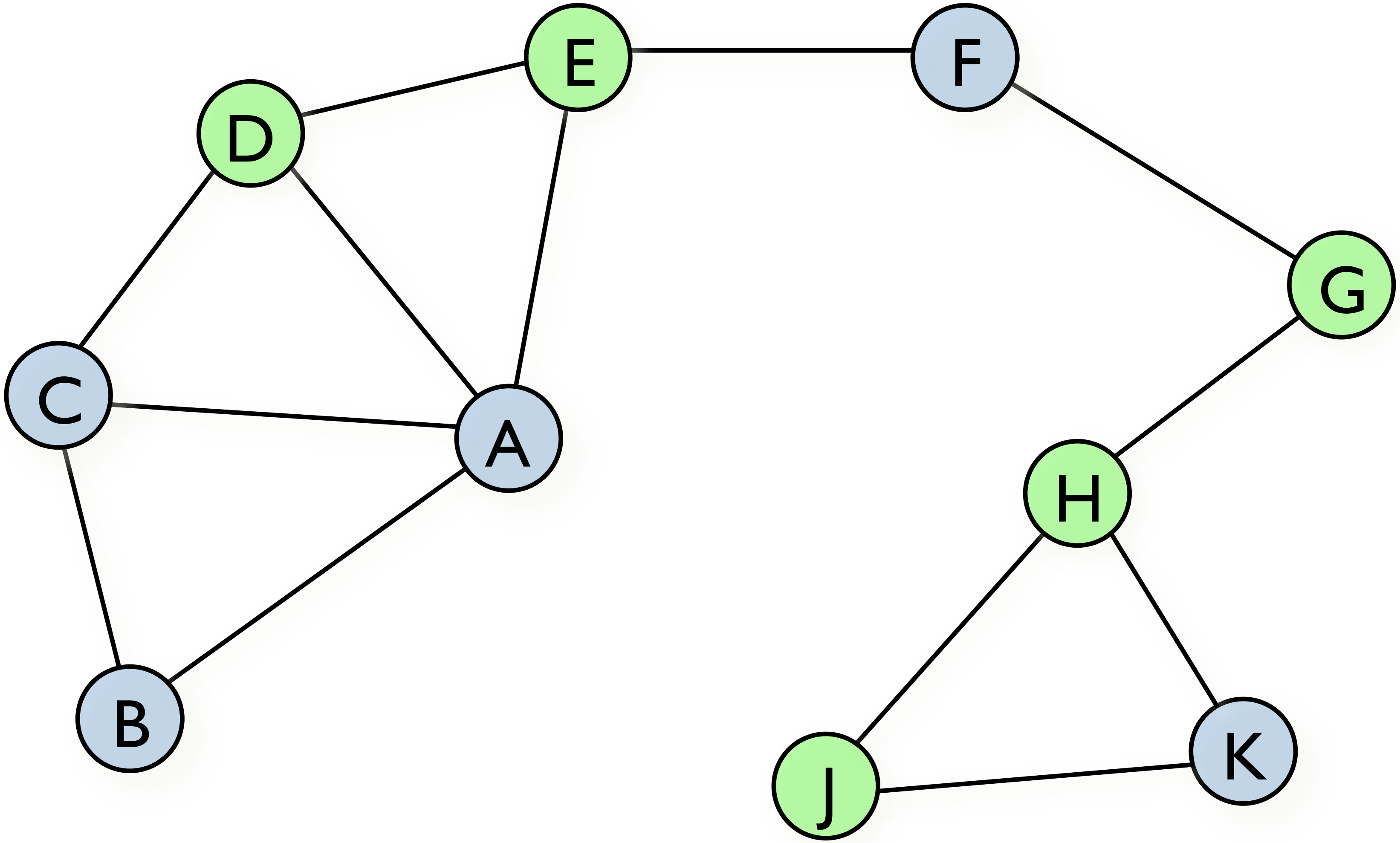
- Agents = nodes in a P2P overlay network
- Each node links to some neighbors (view) in overlay
- Assume:
  - Interaction between neighbors to achive some application task
  - Behavior: Application behavior (i.e. share files or leech files, cooperate or defect)
  - Utility: Evaluated at application level (i.e. number of files downloaded, performace metric)

SLAC algorithm

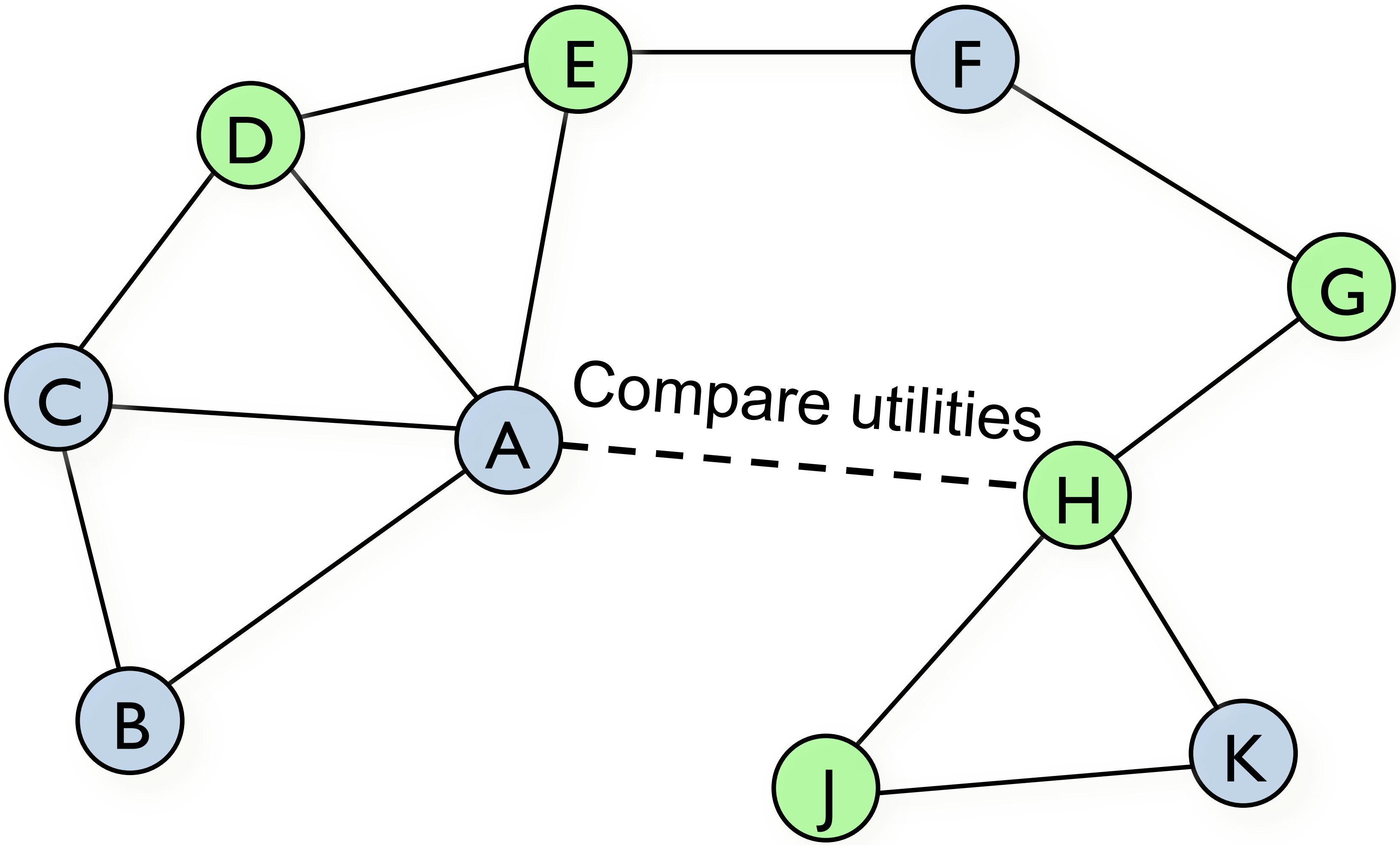
Each node *p* periodically executes the following:

q = SelectRandomPeer()
if utility$_q$ > utility$_p$
    drop all current links
    link to node *q* and copy its strategy and links
    mutate (with low probability) strategy and links
fi

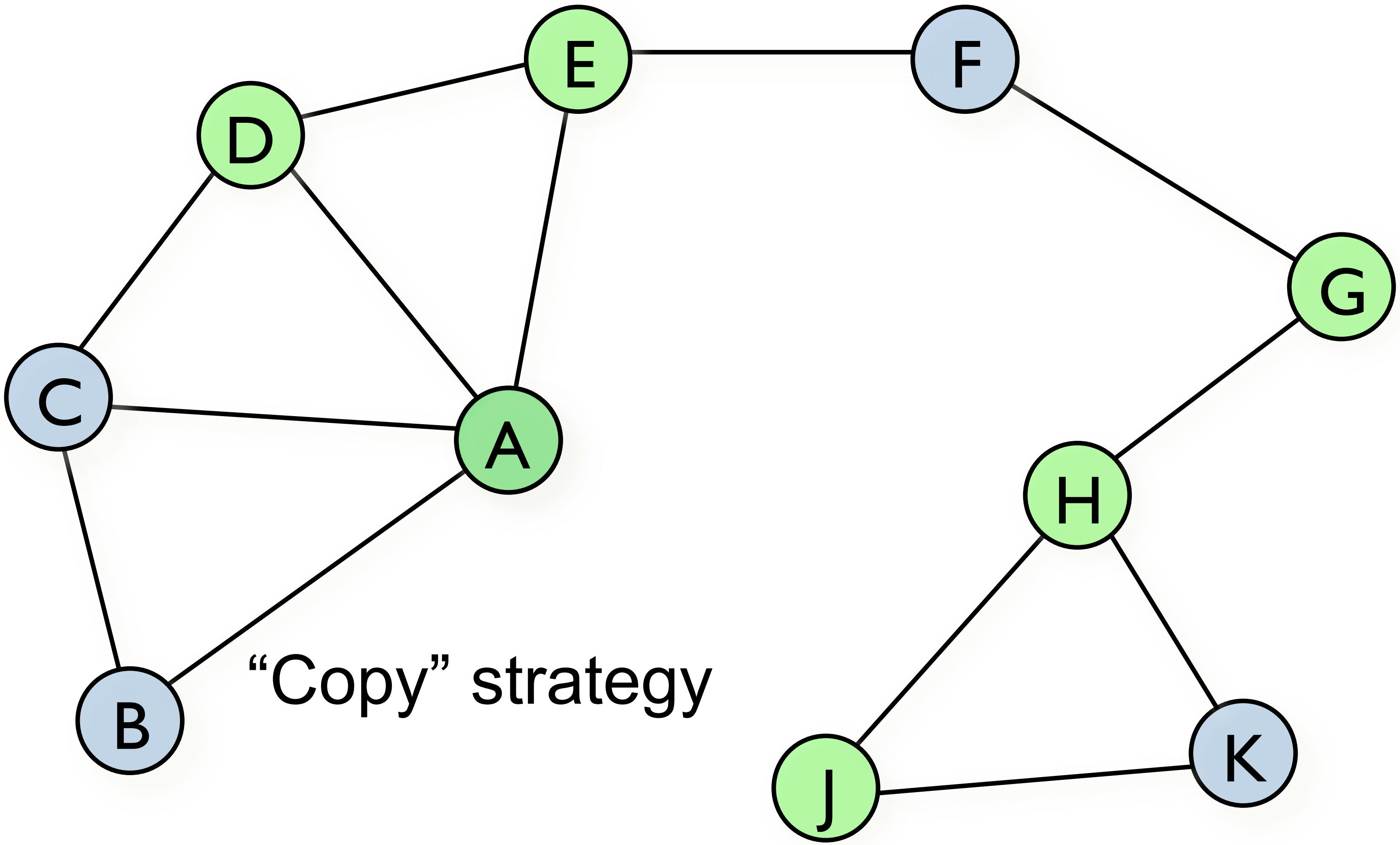
SLAC: "Copy and Rewire"



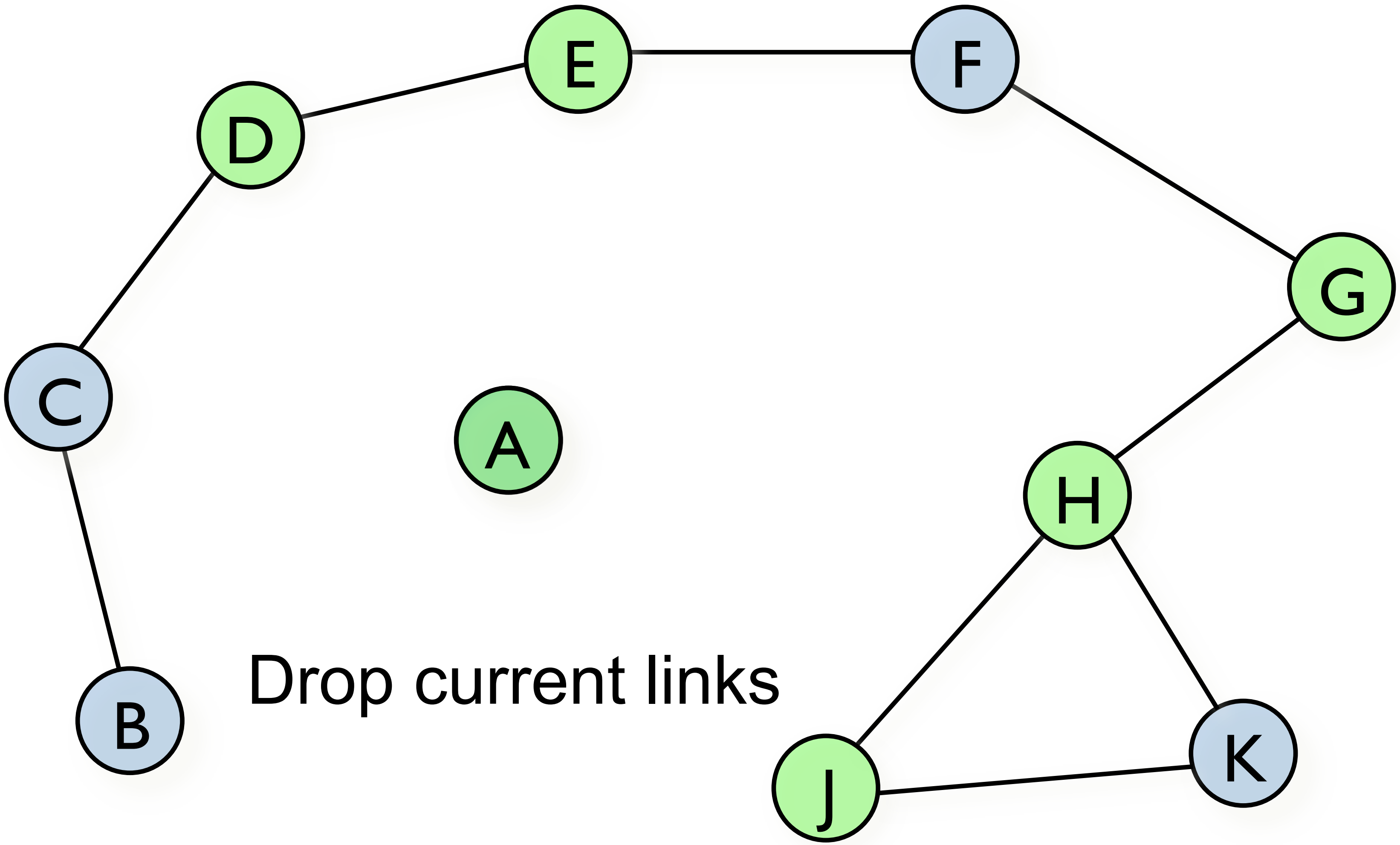
SLAC: “Copy and Rewire”



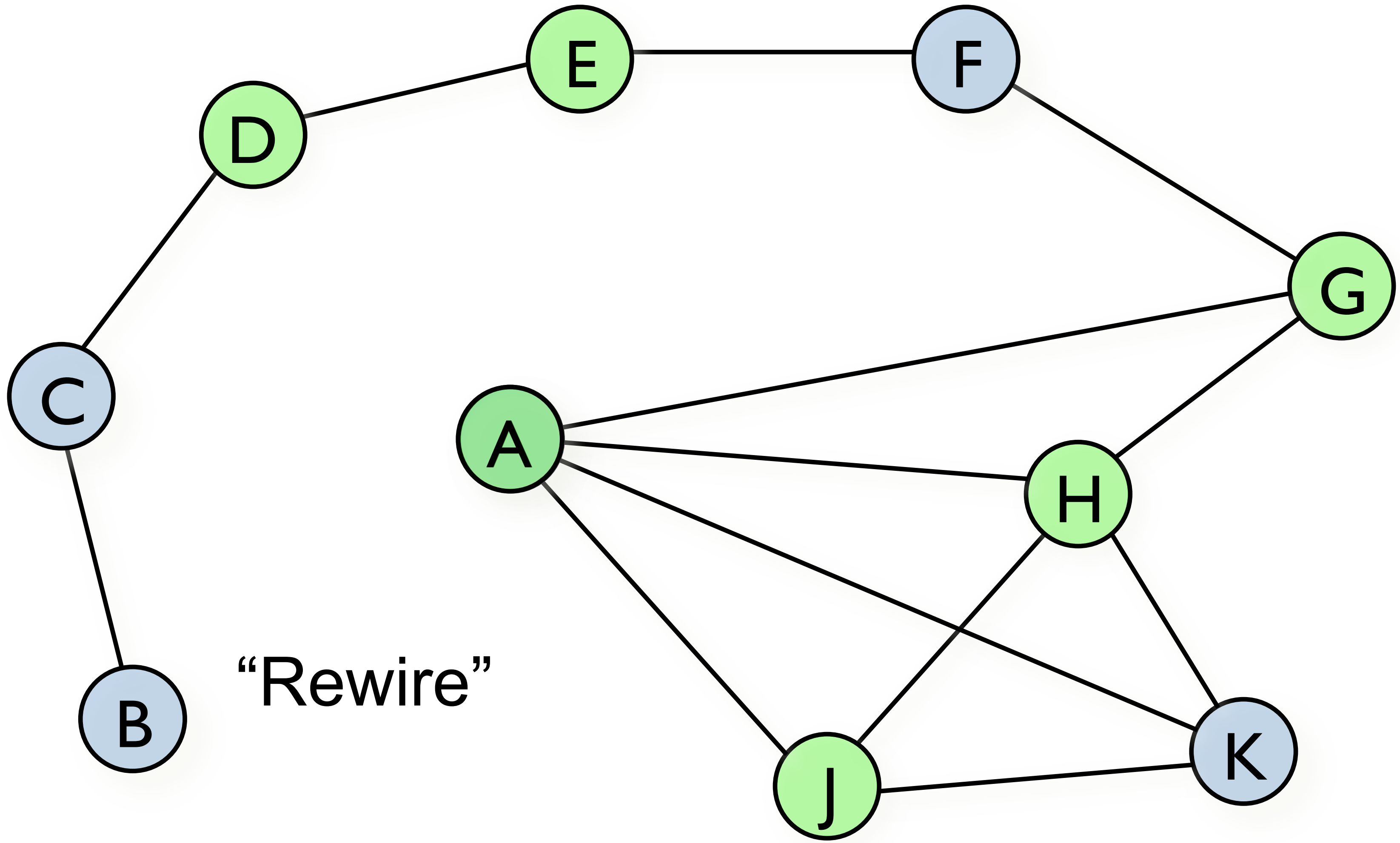
SLAC: "Copy and Rewire"



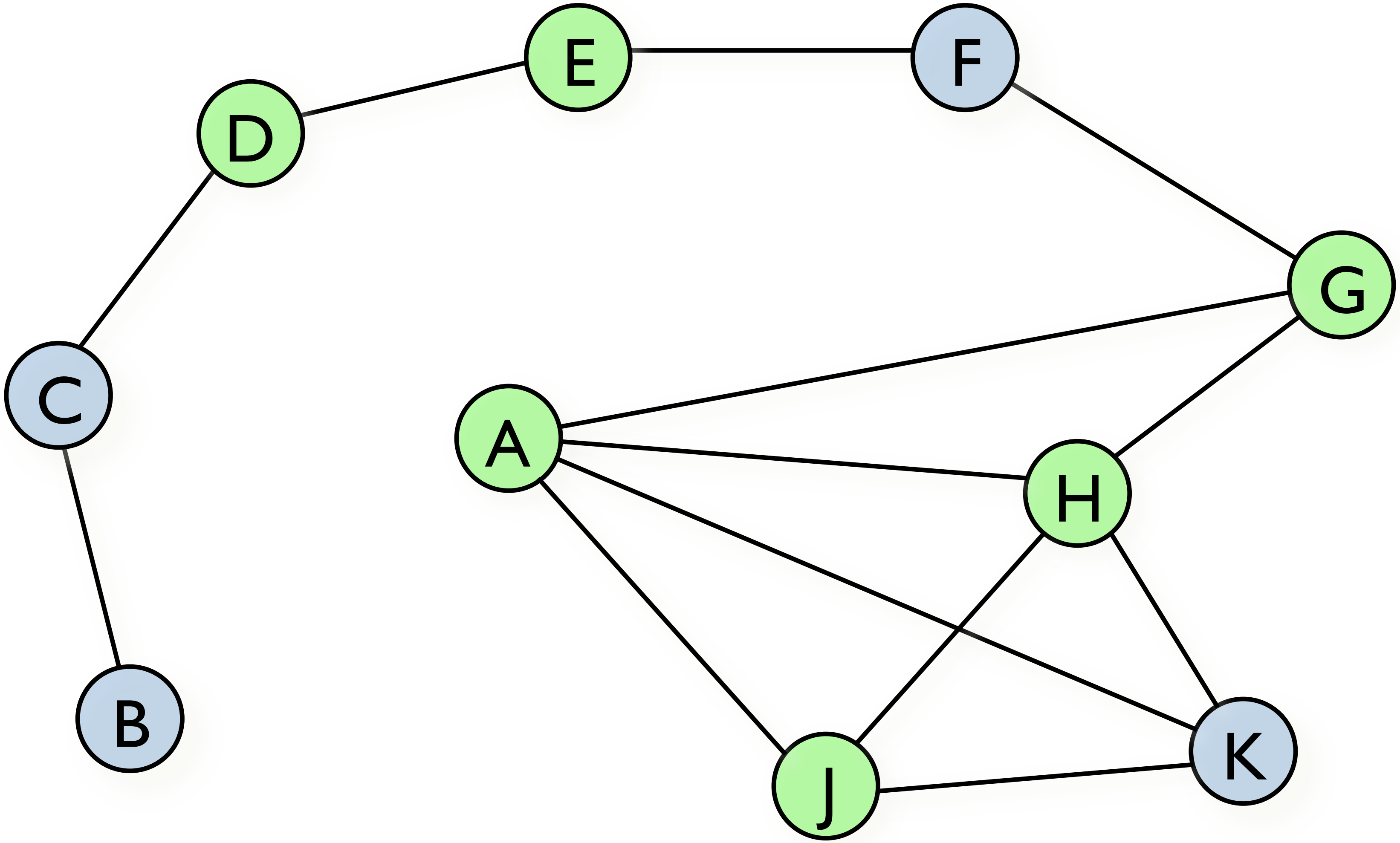
SLAC: "Copy and Rewire"



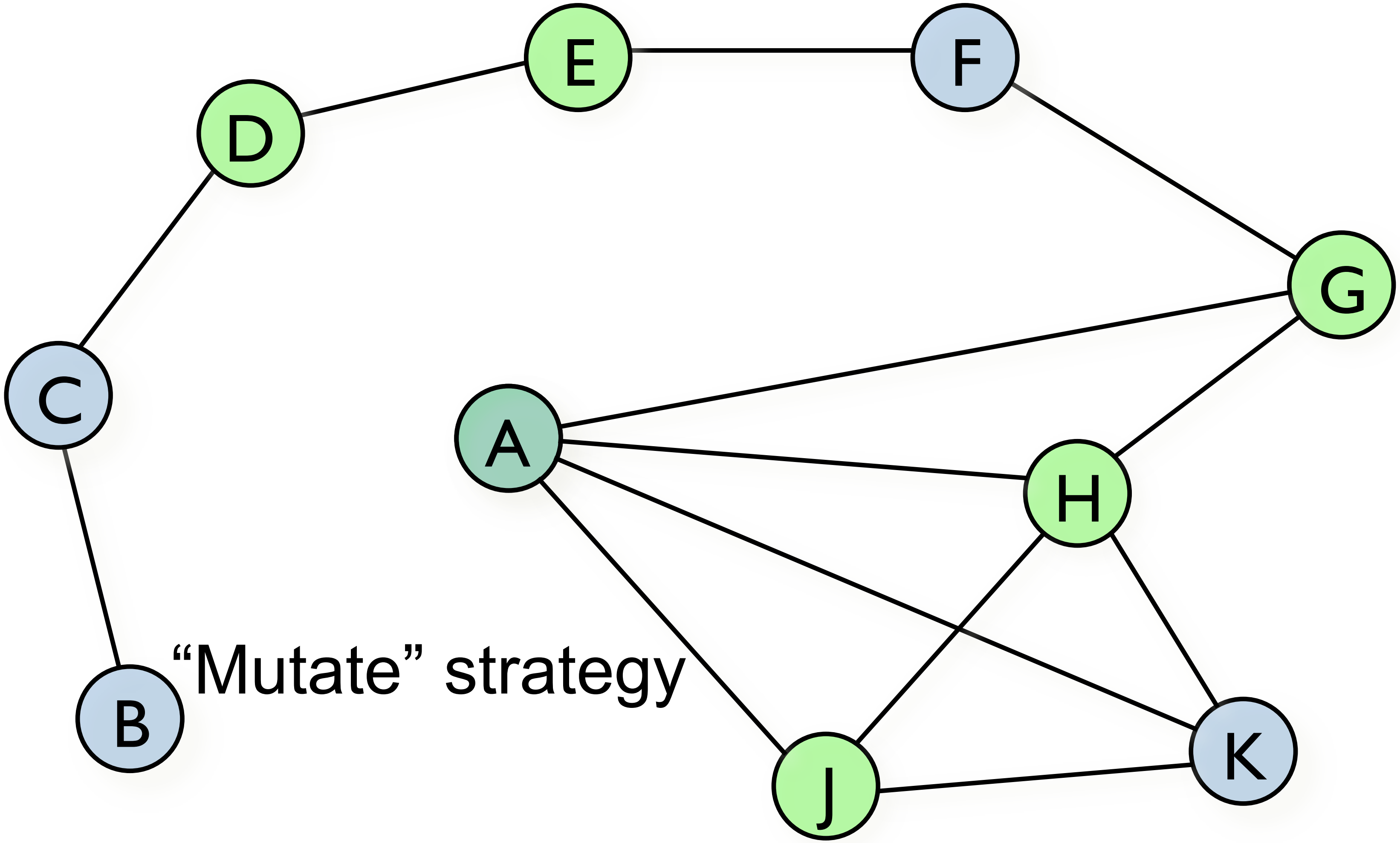
SLAC: “Copy and Rewire”



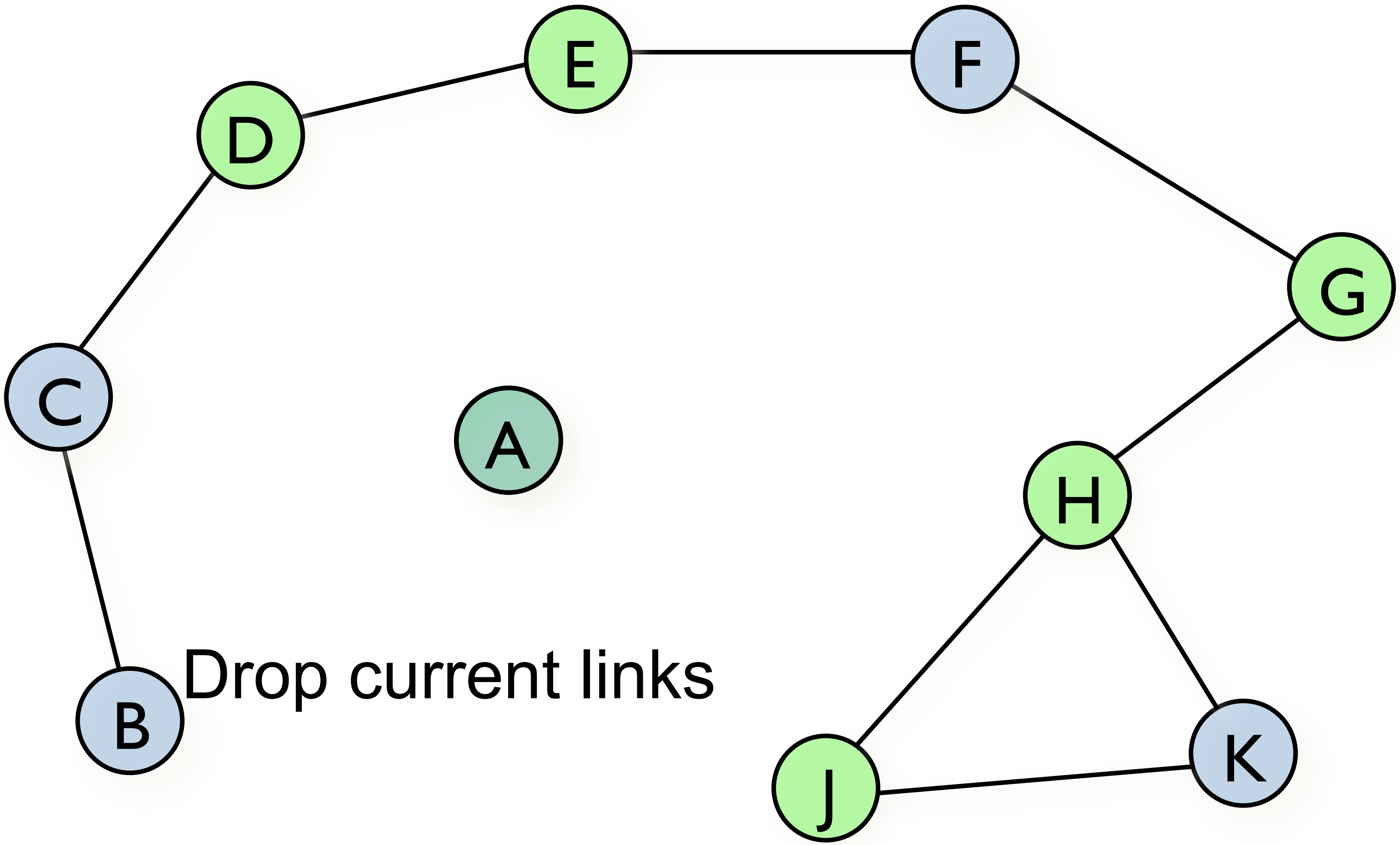
SLAC: “Mutate”



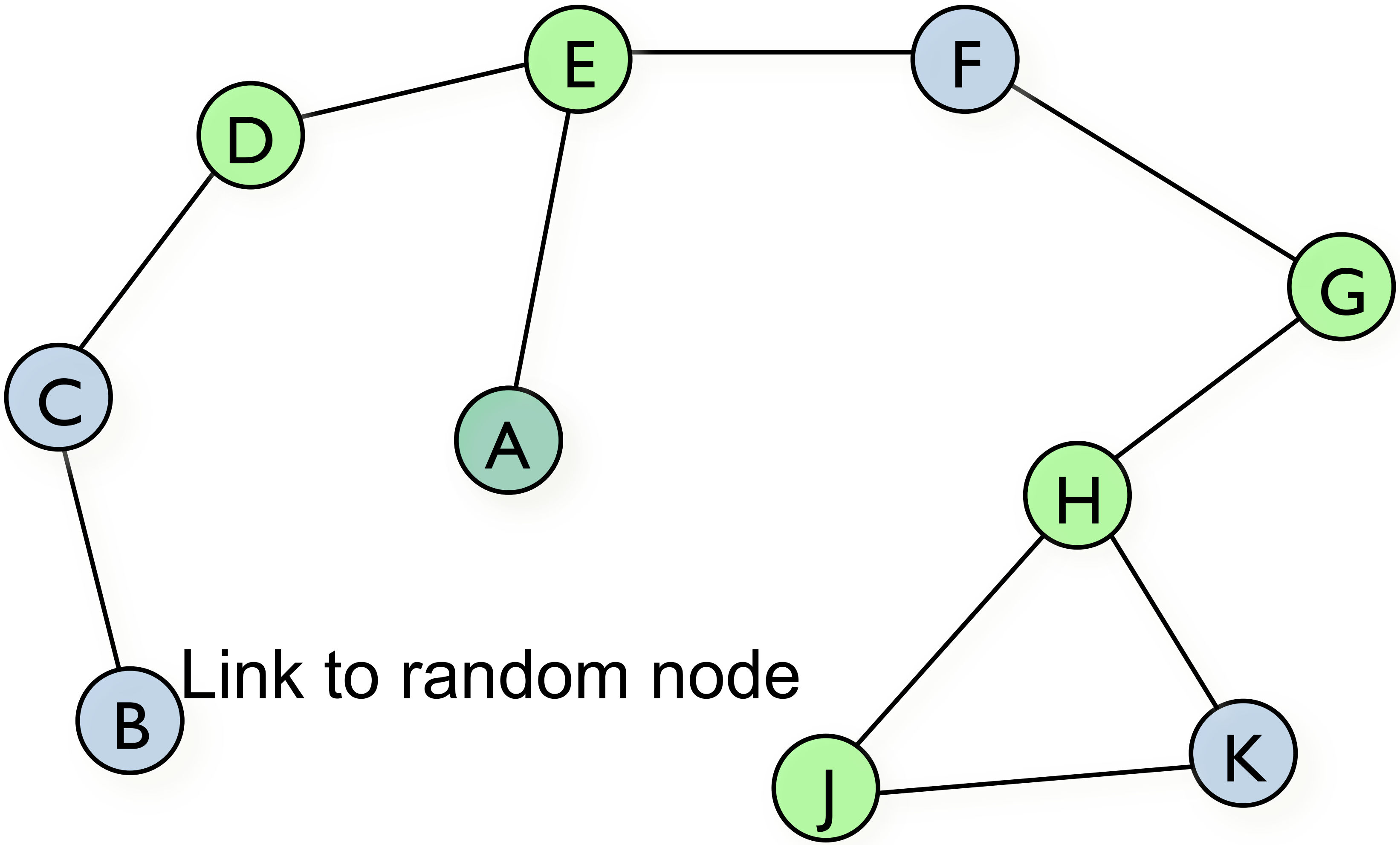
SLAC: “Mutate”



SLAC: “Mutate”



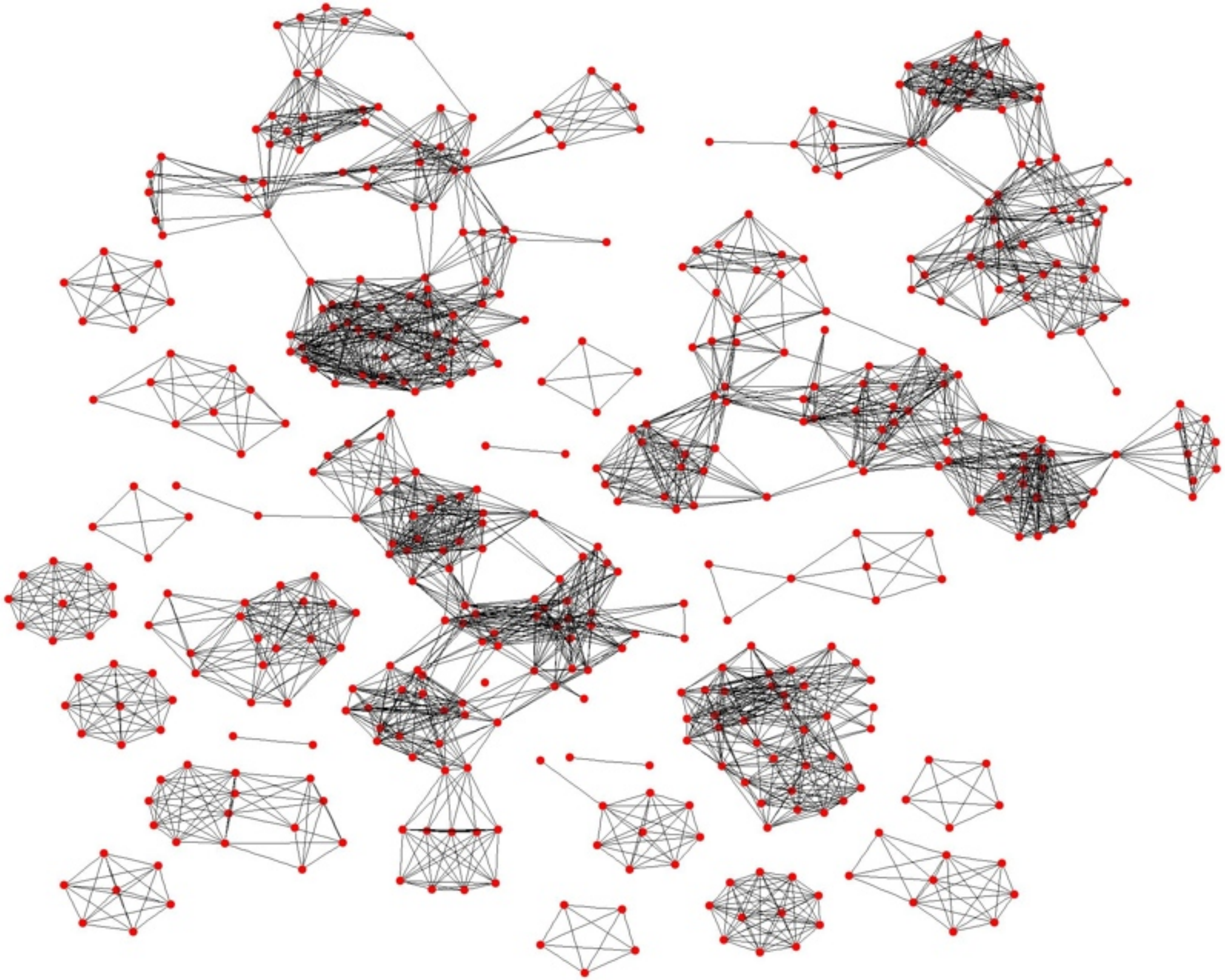
SLAC: "Mutate"



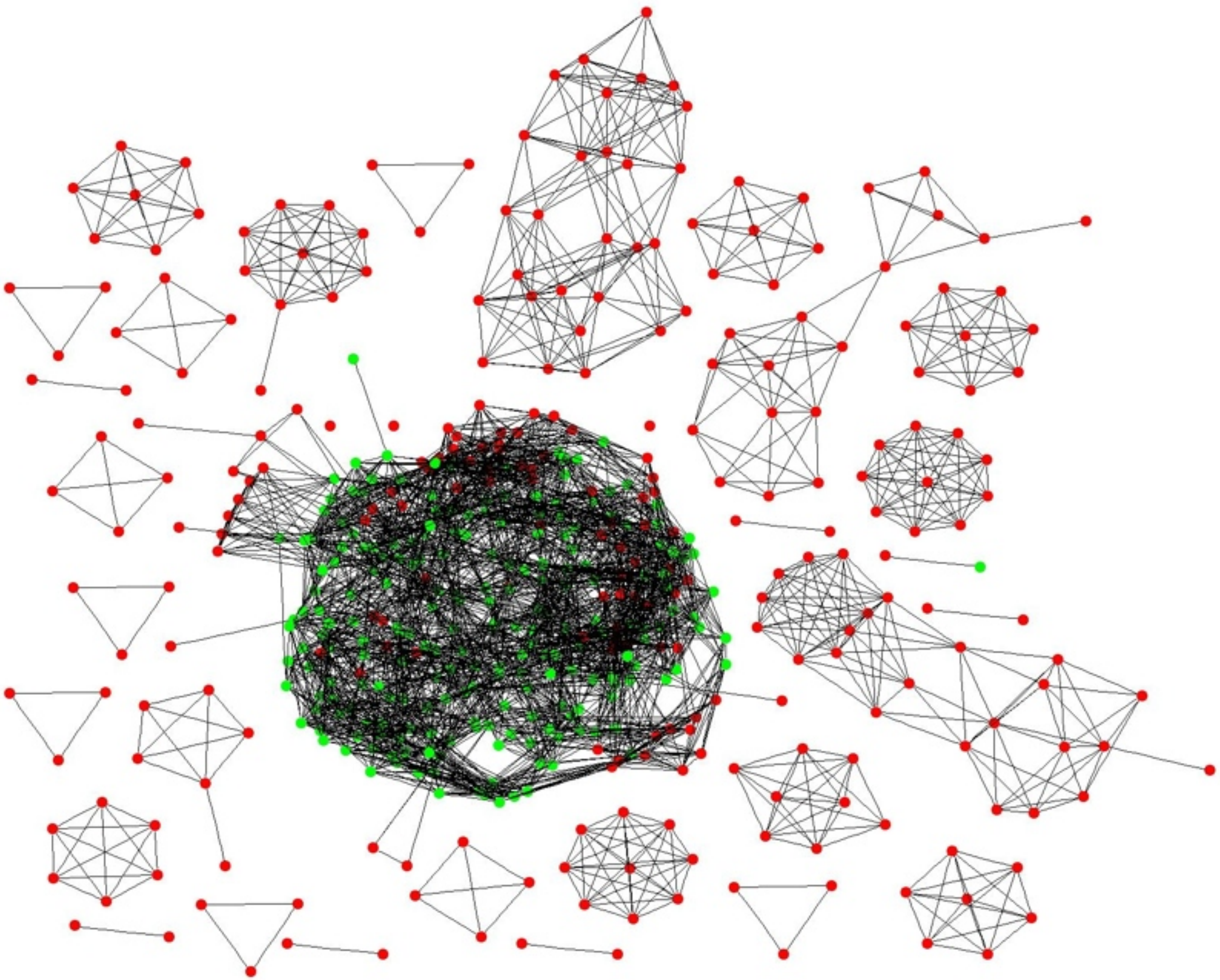
SLAC playing the PD

- We tested SLAC with Prisoner's Dilemma (PD)
  - Captures the conflict between "individual rationality" and "common good"
  - Defection (*D*) leads to higher *individual* utility
  - Cooperation (*C*) leads to higher *global* utility
  - *DC* > *CC* > *DD* > *CD*
- Prisoner's Dilemma in SLAC
  - Nodes play PD with neighbors chosen randomly in the interaction network
  - Only pure strategies (always *C* or always *D*)
  - Strategy mutation: flip current strategy
  - Utility: average payoff achieved

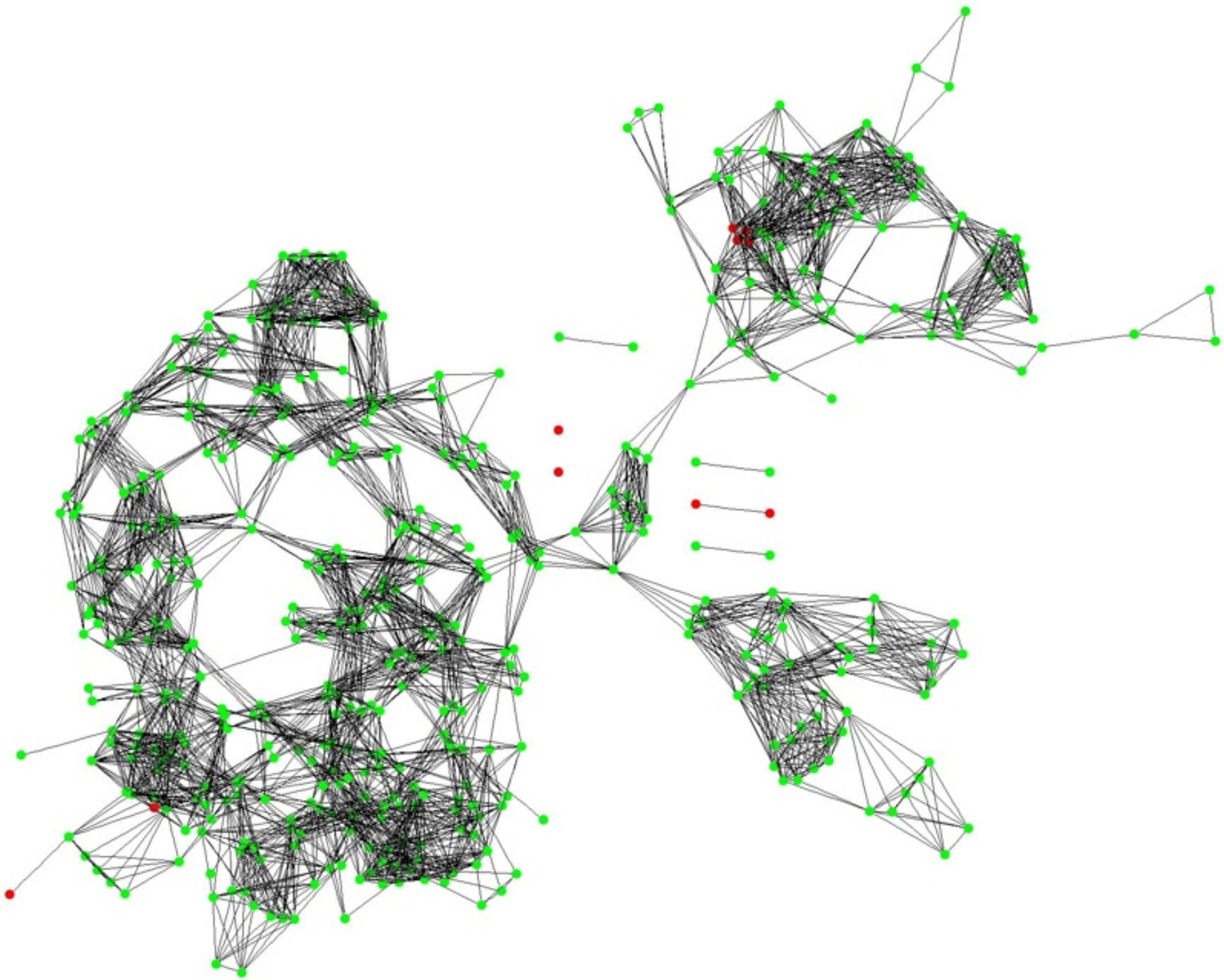
Cycle 180: Small Defect Clusters



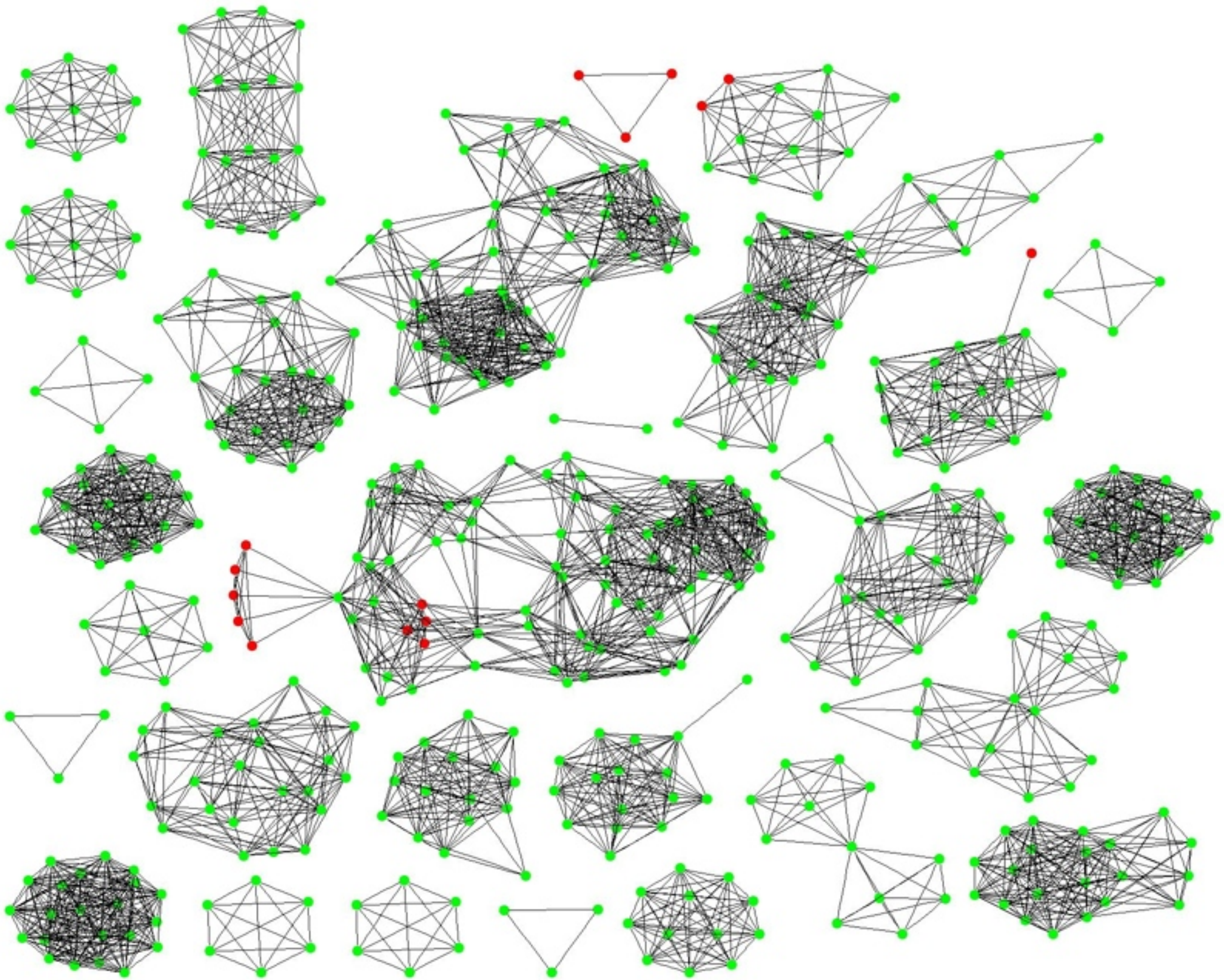
Cycle 220: Cooperation Emerges

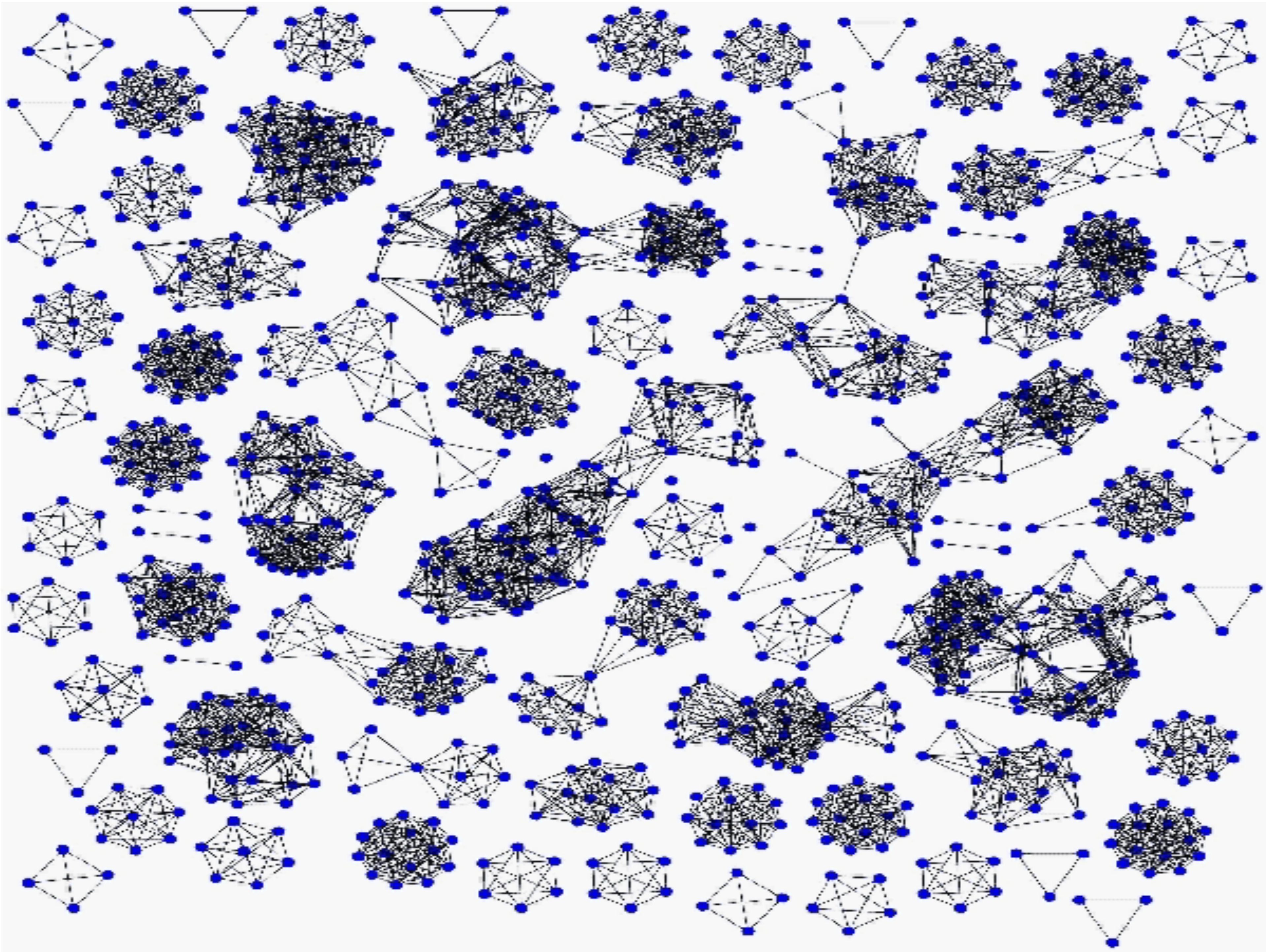


Cycle 230: Coop. Cluster Starts to Break Apart

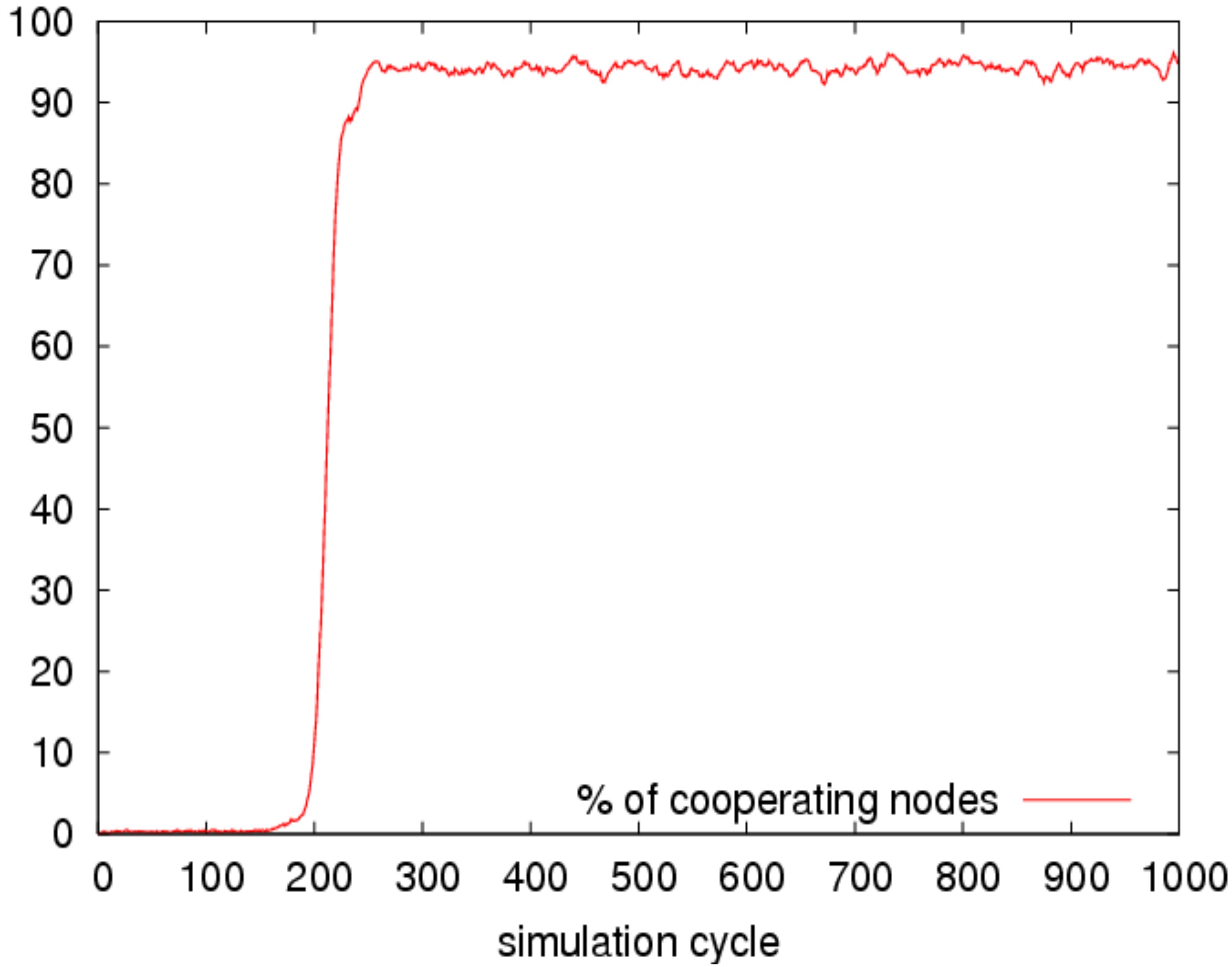


Cycle 300: Defect Nodes Isolated, Small Cooperative Clusters Formed





Cooperation Trend



SLAC Summary

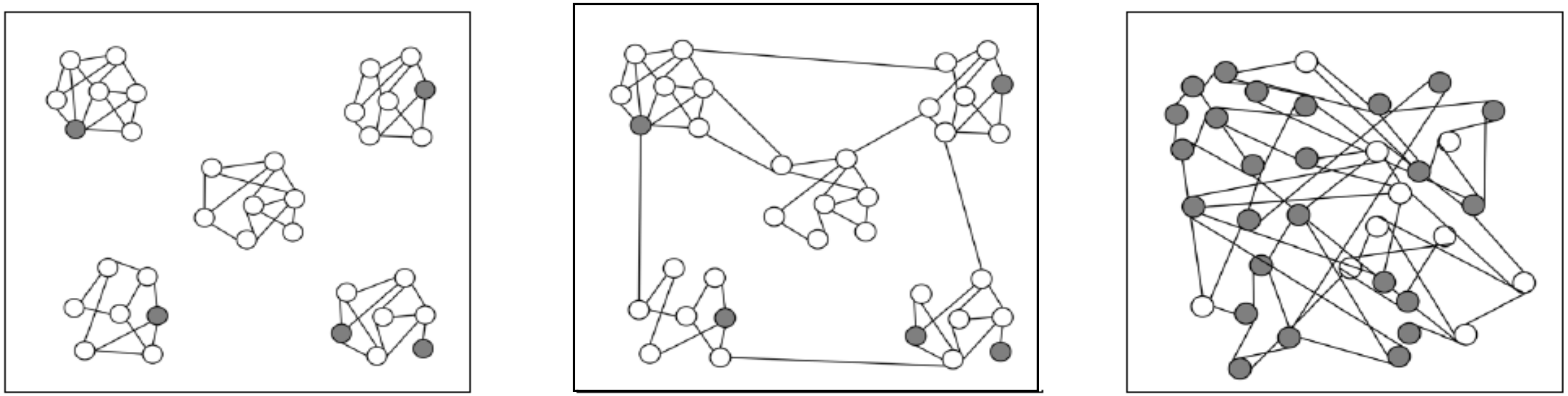
- SLAC produces very high levels of cooperation limits the spread of defection
- Nodes “move” throughout the network to find better neighborhoods
- Group-like selection between clusters
  - Clusters of cooperating nodes grow and persist
  - Defecting nodes tend to become isolated

SLAC and SLACER

- SLAC rewiring mechanism lead to high level of network partitioning
- SLACER: When isolating nodes not all the links are drop. Each link is dropped with given probability *W*
- Parameter *W* represents a tradeoff between network randomness and cooperation level
  - *W*=1: high cooperation, high partitioning
  - *W*=0.9: high cooperation, small world like topology
  - Low *W*: low cooperation, random like topology

SLAC and SLACER

W=1 W=0.9 W=0.3



SLAC SLACER

As W is increased (probability of dropping a link when moving) then the network becomes more random and cooperation reduces. Intermeidate points give small-world fully connected networks

SLAC and SLACER

- We applied variants of SLAC and SLACER in P2P applications:
- File-sharing
- Content replication for webservers
- Job sharing requiring specialisation in the clusters in addition to cooperation

Elinor Ostrom 1990

Ostrom identifies eight "design principles" of stable local common pool resource management:

1. Clearly defined boundaries (effective exclusion of external unentitled parties);
2. Rules regarding the appropriation and provision of common resources are adapted to local conditions;
3. Collective-choice arrangements allow most resource appropriators to participate in the decision-making process;
4. Effective monitoring by monitors who are part of or accountable to the appropriators;
5. There is a scale of graduated sanctions for resource appropriators who violate community rules;
6. Mechanisms of conflict resolution are cheap and of easy access;
7. The self-determination of the community is recognized by higher-level authorities;
8. In the case of larger common-pool resources: organization in the form of multiple layers of nested enterprises, with small local CPRs at the base level.

User Models

- We need realistic models of how users behave when embedded within given ICT systems
- A priori theoretical models tend not work – users rarely behave “rationally” in the sense of maximising some simple utility
- Empirical measurements suggest its complex – heterogeneous, adaptive, but progress can be made
- Need large-scale deployments / measurements – an empirical / experimental approach

Rawls' "veil of ignorance" approach

- assume we wish to specify the kind of society that is just and good
- but we stand outside the society and don't know what role we ourselves would play
  - we are ignorant of what endowments, knowledge, capacities and position we would hold
- what rules / norms would we accept as just and fair? i.e. what would we accept as "collective good"

Designing for common good

- We wish to specify the requirements of a system that will structure interaction between peers
- the protocol could run on diverse devices with diverse goals, capacities and user behaviour
- but we need 1 billion users of the system to make it a success (and get rich)
- What collective goals will we define such that many different devices and users would accept and run it?
    - “do no evil”? or “make the world a better place”? or “from each according to his abilities to each according to his need”?